DATABASE AND CHILL

Ariele Silva

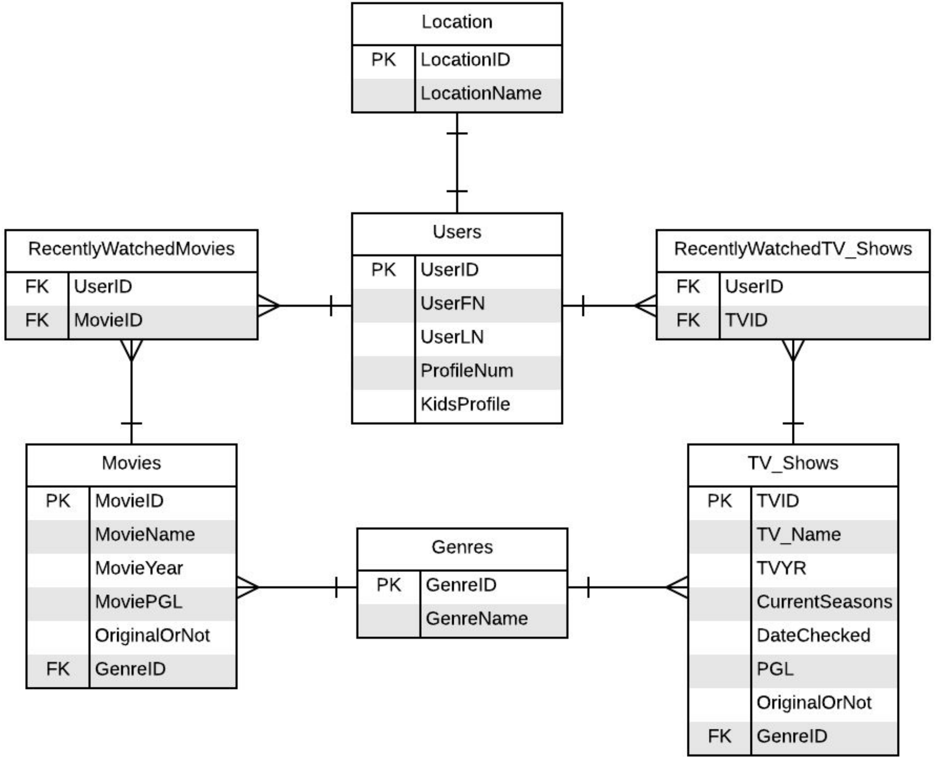
THE NETFLIX DATABASE

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This database was created for the purpose of keeping track of Netflix's user's streaming. By tracking each user's watching habits we can better customize suggestions and see what shows and movies are currently trending. We can also access each user's recently watched history whether it be movies or TV shows. From there each user in the database can even search by genre whether it be a move or a TV show, to get a list of what is available to watch. It gives the users many options and it gives Netflix the data and information needed to know which TV shows or movies are the most watched and how many TV shows or movies each user watches, etc.

ER DIAGRAM



Location	
PK	LocationID
	LocationName

RecentlyWatchedMovies	
FK	UserID
FK	MovieID

Users	
PK	UserID
	UserFN
	UserLN
	ProfileNum
	KidsProfile

RecentlyWatchedTV_Shows	
FK	UserID
FK	TVID

Movies	
PK	MovieID
	MovieName
	MovieYear
	MoviePGL
	OriginalOrNot
FK	GenreID

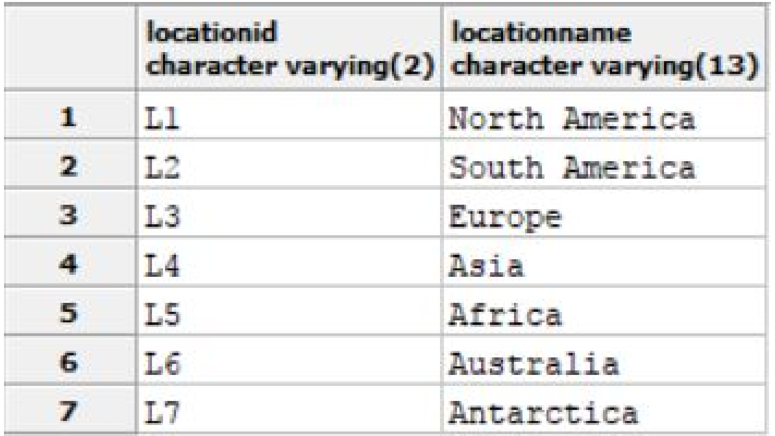
Genres	
PK	GenreID
	GenreName

TV_Shows	
PK	TVID
	TV_Name
	TVYR
	CurrentSeasons
	DateChecked
	PGL
	OriginalOrNot
FK	GenreID

LOCATION TABLE

This table contains the locations in which Netflix streaming is available.

```
DROP TABLE IF EXISTS Location;
CREATE TABLE Location (
    LocationID VARCHAR(2),
    LocationName VARCHAR(13),
        PRIMARY KEY(LocationID)
);
```



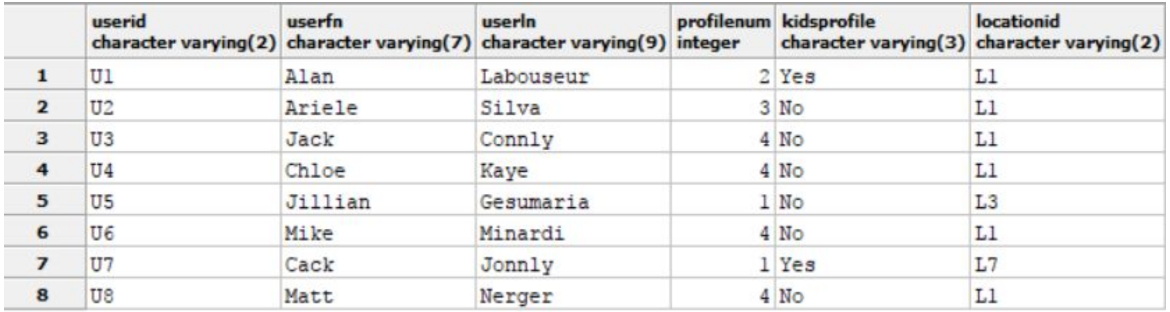
	locationid character varying(2)	locationname character varying(13)
1	L1	North America
2	L2	South America
3	L3	Europe
4	L4	Asia
5	L5	Africa
6	L6	Australia
7	L7	Antarctica

Functional Dependencies: **LocationID** → LocationName

USERS TABLE

This table is made up of all the users for Netflix’s streaming service. It contains the user’s account information like the user’s ID, their first name, last name and location. It also has the amount of profiles on each account and whether or not there is a kid friendly profile registered on the account.

```
DROP TABLE IF EXISTS Users;
CREATE TABLE Users (
   UserID VARCHAR(2),
   UserFN VARCHAR(7),
   UserLN VARCHAR(9),
   ProfileNum INT,
   KidsProfile VARCHAR(3),
   LocationID VARCHAR(2),
        PRIMARY KEY(UserID),
        FOREIGN KEY (LocationID)
REFERENCES Location(LocationID)
);
```



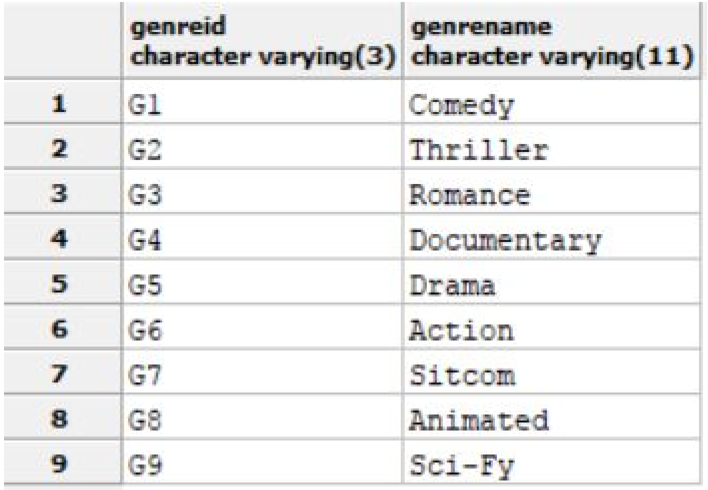
	userid character varying(2)	userfn character varying(7)	userln character varying(9)	profilenum integer	kidsprofile character varying(3)	locationid character varying(2)
1	U1	Alan	Labouseur	2	Yes	L1
2	U2	Ariele	Silva	3	No	L1
3	U3	Jack	Connly	4	No	L1
4	U4	Chloe	Kaye	4	No	L1
5	U5	Jillian	Gesumaria	1	No	L3
6	U6	Mike	Minardi	4	No	L1
7	U7	Cack	Jonnly	1	Yes	L7
8	U8	Matt	Nerger	4	No	L1

Functional Dependencies: **UsersID** → UserFN, UserLN, ProfileNum, KidsProfile

GENRES TABLE

All the genres that Netflix uses to categorize it’s content will be found in this table.

```
DROP TABLE IF EXISTS Genres;
CREATE TABLE Genres (
    GenreID VARCHAR(3),
    GenreName VARCHAR(11),
       PRIMARY KEY(GenreID)
);
```

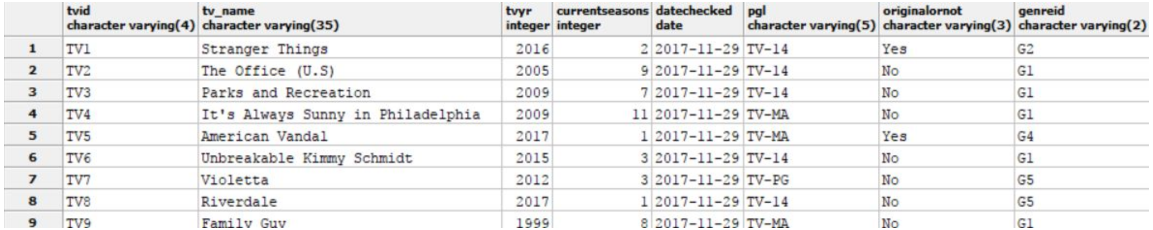


	genreid character varying(3)	genrename character varying(11)
1	G1	Comedy
2	G2	Thriller
3	G3	Romance
4	G4	Documentary
5	G5	Drama
6	G6	Action
7	G7	Sitcom
8	G8	Animated
9	G9	Sci-Fy

Functional Dependencies: **GenresID** → GenreName

TV SHOWS TABLE

The TV Shows table contains all the show's in Netflix's streaming service. The fields are the TV Show's ID, the name of the show, the release date of the show, the number of season's Netflix has posted, the date checked to make sure the season's count is up to date, the parental guidelines rating, whether or not it is a Netflix Original, and the type of genre.



	tvid character varying(4)	tv_name character varying(35)	tvyr integer	currentseasons integer	datechecked date	pgl character varying(5)	originalornot character varying(3)	genreid character varying(2)
1	TV1	Stranger Things	2016	2	2017-11-29	TV-14	Yes	G2
2	TV2	The Office (U.S)	2005	9	2017-11-29	TV-14	No	G1
3	TV3	Parks and Recreation	2009	7	2017-11-29	TV-14	No	G1
4	TV4	It's Always Sunny in Philadelphia	2009	11	2017-11-29	TV-MA	No	G1
5	TV5	American Vandal	2017	1	2017-11-29	TV-MA	Yes	G4
6	TV6	Unbreakable Kimmy Schmidt	2015	3	2017-11-29	TV-14	No	G1
7	TV7	Violetta	2012	3	2017-11-29	TV-PG	No	G5
8	TV8	Riverdale	2017	1	2017-11-29	TV-14	No	G5
9	TV9	Family Guy	1999	8	2017-11-29	TV-MA	No	G1

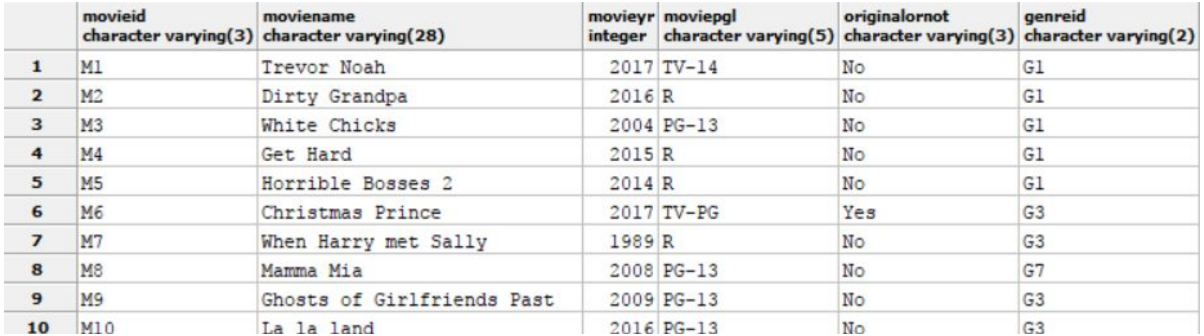
Functional Dependencies: **TVID** → TV_Name, TVYR, CurrentSeasons, DateChecked, PGL, OriginalOrNot, GenreID

```
DROP TABLE IF EXISTS TV_Shows;
CREATE TABLE TV_Shows (
   TVID VARCHAR(4),
   TV_Name VARCHAR(35),
   TVYR INT,
   CurrentSeasons INT,
   DateChecked DATE,
   PGL VARCHAR(5),
   OriginalOrNot VARCHAR(3),
   GenreID VARCHAR(2),
          PRIMARY KEY(TVID),
          FOREIGN KEY (GenreID) REFERENCES
Genres(GenreID)
);
```

MOVIES TABLE

The Movies table contains all the movies in Netflix's streaming service. The fields are the Movie ID, the name of the movie, the movies release date, the parental guidelines rating, whether or not it is a Netflix Original, and the type of genre.

```
DROP TABLE IF EXISTS Movies;
CREATE TABLE Movies (
    MovieID VARCHAR(3),
    MovieName VARCHAR(28),
    MovieYR INT,
    MoviePGL VARCHAR(5),
    OriginalOrNot VARCHAR(3),
    GenreID VARCHAR(2),
         PRIMARY KEY (MovieID),
         FOREIGN KEY (GenreID)
REFERENCES Genres(GenreID)
);
```



	movieid	moviename	movieyr	moviepgl	originalornot	genreid
	character varying(3)	character varying(28)	integer	character varying(5)	character varying(3)	character varying(2)
1	M1	Trevor Noah	2017	TV-14	No	G1
2	M2	Dirty Grandpa	2016	R	No	G1
3	M3	White Chicks	2004	PG-13	No	G1
4	M4	Get Hard	2015	R	No	G1
5	M5	Horrible Bosses 2	2014	R	No	G1
6	M6	Christmas Prince	2017	TV-PG	Yes	G3
7	M7	When Harry met Sally	1989	R	No	G3
8	M8	Mamma Mia	2008	PG-13	No	G7
9	M9	Ghosts of Girlfriends Past	2009	PG-13	No	G3
10	M10	La la land	2016	PG-13	No	G3

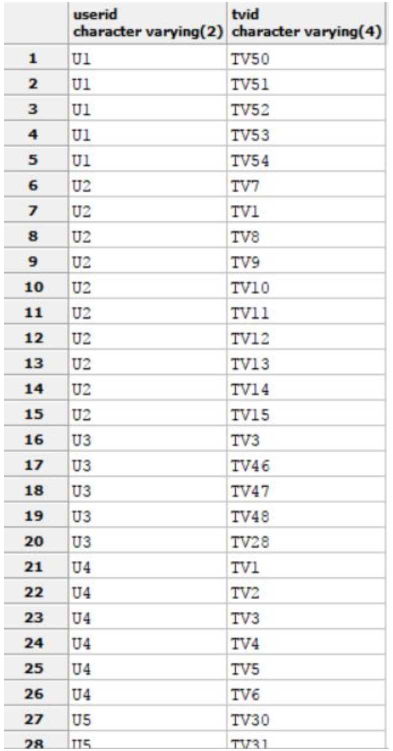
Functional Dependencies: **MovieID** → MovieName, MovieYR, MoviePGL, OriginalOrNot, GenreID

RECENTLY WATCHED TV SHOWS

This table consists of the user's ID and the TV shows ID. It shows all of the shows that the each user has recently watched.

```
DROP TABLE IF EXISTS RecentlyWatchedTVShows;
CREATE TABLE RecentlyWatchedTVShows (
    UserID VARCHAR(2),
    TVID VARCHAR(4),
          FOREIGN KEY (UserID) REFERENCES Users(UserID),
          FOREIGN KEY (TVID) REFERENCES TV_Shows(TVID)
);
```

Functional Dependencies: **UsersID** → UserFN, UserLN, ProfileNum, KidsProfile
TVID → TV_Name, TVYR, CurrentSeasons, DateChecked, PGL, OriginalOrNot, GenreID



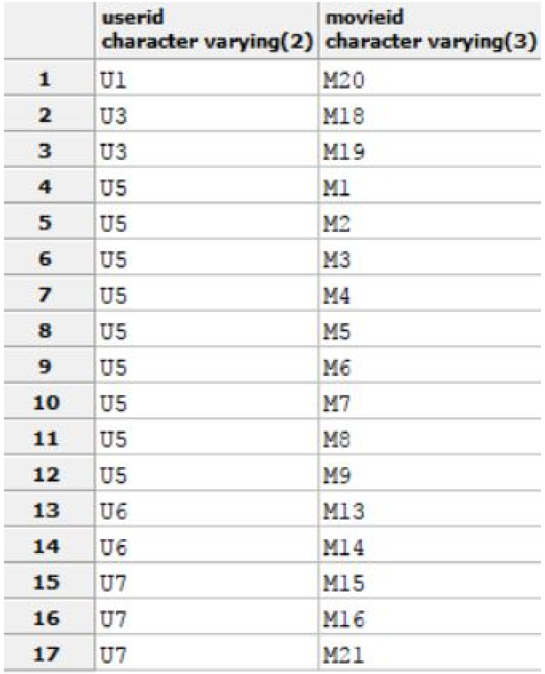
	userid character varying(2)	tvid character varying(4)
1	U1	TV50
2	U1	TV51
3	U1	TV52
4	U1	TV53
5	U1	TV54
6	U2	TV7
7	U2	TV1
8	U2	TV8
9	U2	TV9
10	U2	TV10
11	U2	TV11
12	U2	TV12
13	U2	TV13
14	U2	TV14
15	U2	TV15
16	U3	TV3
17	U3	TV46
18	U3	TV47
19	U3	TV48
20	U3	TV28
21	U4	TV1
22	U4	TV2
23	U4	TV3
24	U4	TV4
25	U4	TV5
26	U4	TV6
27	U5	TV30
28	U5	TV31

RECENTLY WATCHED MOVIES

This table consists of the user's ID and the TV shows ID. It shows all of the shows that the each user has recently watched.

```
DROP TABLE IF EXISTS RecentlyWatchedMovies;
CREATE TABLE RecentlyWatchedMovies (
    UserID VARCHAR(2),
    MovieID VARCHAR(3),
         FOREIGN KEY (UserID) REFERENCES Users(UserID),
         FOREIGN KEY (MovieID) REFERENCES Movies(MovieID)
);
```

Functional Dependencies: **UsersID** → UserFN, UserLN, ProfileNum, KidsProfile
MovieID → MovieName, MovieYR, MoviePGL, OriginalOrNot, GenreID



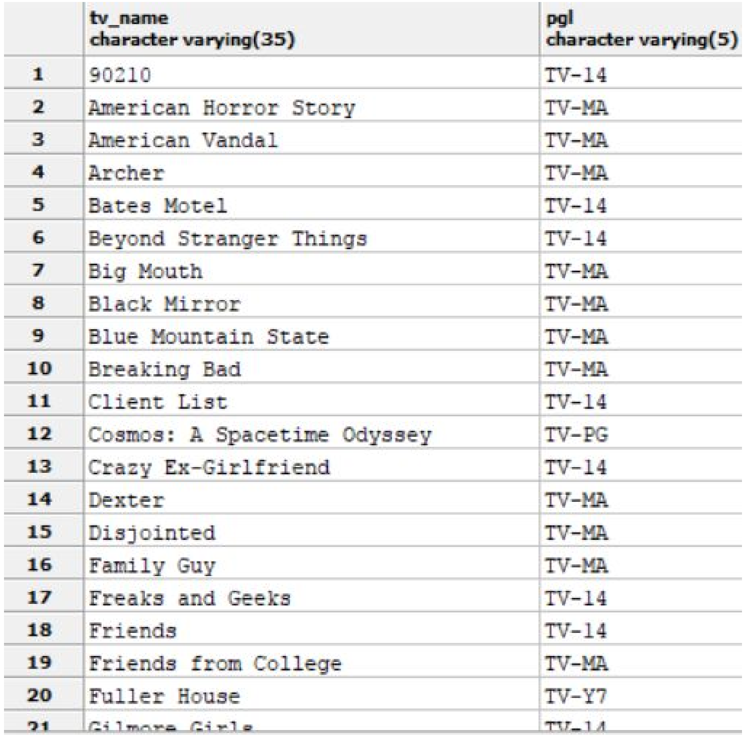
	userid character varying(2)	movieid character varying(3)
1	U1	M20
2	U3	M18
3	U3	M19
4	U5	M1
5	U5	M2
6	U5	M3
7	U5	M4
8	U5	M5
9	U5	M6
10	U5	M7
11	U5	M8
12	U5	M9
13	U6	M13
14	U6	M14
15	U7	M15
16	U7	M16
17	U7	M21

TVSHOWS VIEW

This view gets all the TV shows names in alphabetical order and their parental guidelines rating.

```
CREATE OR REPLACE VIEW TVSHOWS
    AS SELECT DISTINCT
TV_Shows.TV_Name, TV_Shows.PGL
    FROM TV_Shows;

 SELECT * FROM TVSHOWS
 ORDER BY TV_NAME ASC;
```



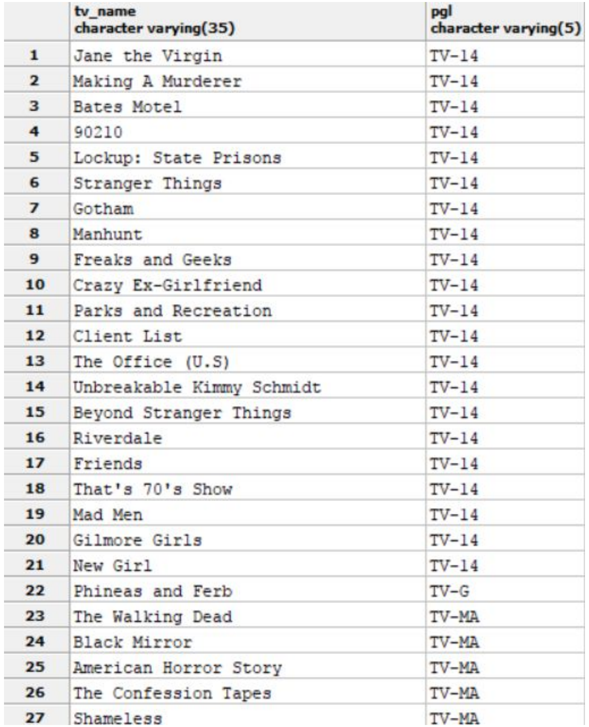
	tv_name character varying(35)	pgl character varying(5)
1	90210	TV-14
2	American Horror Story	TV-MA
3	American Vandal	TV-MA
4	Archer	TV-MA
5	Bates Motel	TV-14
6	Beyond Stranger Things	TV-14
7	Big Mouth	TV-MA
8	Black Mirror	TV-MA
9	Blue Mountain State	TV-MA
10	Breaking Bad	TV-MA
11	Client List	TV-14
12	Cosmos: A Spacetime Odyssey	TV-PG
13	Crazy Ex-Girlfriend	TV-14
14	Dexter	TV-MA
15	Disjointed	TV-MA
16	Family Guy	TV-MA
17	Freaks and Geeks	TV-14
18	Friends	TV-14
19	Friends from College	TV-MA
20	Fuller House	TV-Y7
21	Gilmore Girls	TV-14

TVSHOWS RATING VIEW

This view gets all the parental guidelines ratings in groupings and their TV shows names.

```
CREATE OR REPLACE VIEW TVSHOWSRATING
   AS SELECT DISTINCT
TV_Shows.PGL,TV_Shows.TV_Name
   FROM TV_Shows;

SELECT * FROM TVSHOWS
ORDER BY PGL ASC;
```



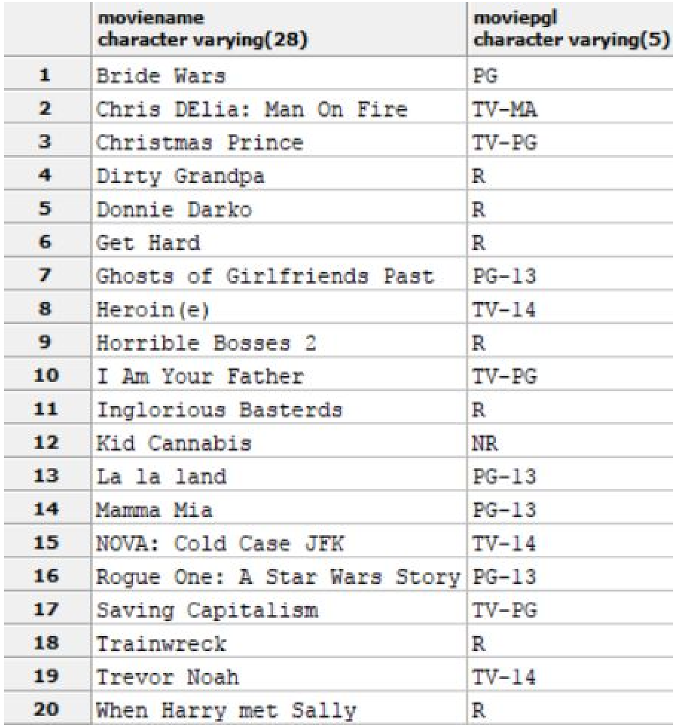
	tv_name	pgl
	character varying(35)	character varying(5)
1	Jane the Virgin	TV-14
2	Making A Murderer	TV-14
3	Bates Motel	TV-14
4	90210	TV-14
5	Lockup: State Prisons	TV-14
6	Stranger Things	TV-14
7	Gotham	TV-14
8	Manhunt	TV-14
9	Freaks and Geeks	TV-14
10	Crazy Ex-Girlfriend	TV-14
11	Parks and Recreation	TV-14
12	Client List	TV-14
13	The Office (U.S)	TV-14
14	Unbreakable Kimmy Schmidt	TV-14
15	Beyond Stranger Things	TV-14
16	Riverdale	TV-14
17	Friends	TV-14
18	That's 70's Show	TV-14
19	Mad Men	TV-14
20	Gilmore Girls	TV-14
21	New Girl	TV-14
22	Phineas and Ferb	TV-G
23	The Walking Dead	TV-MA
24	Black Mirror	TV-MA
25	American Horror Story	TV-MA
26	The Confession Tapes	TV-MA
27	Shameless	TV-MA

MOVIES VIEW

This view gets all the movies Netflix is streaming in alphabetical order and their parental guidelines ratings.

```
CREATE OR REPLACE VIEW MOVIE
    AS SELECT DISTINCT Movies.MovieName,
Movies.MoviePGL
    FROM Movies;

 SELECT * FROM MOVIE
 ORDER BY MovieName ASC;
```



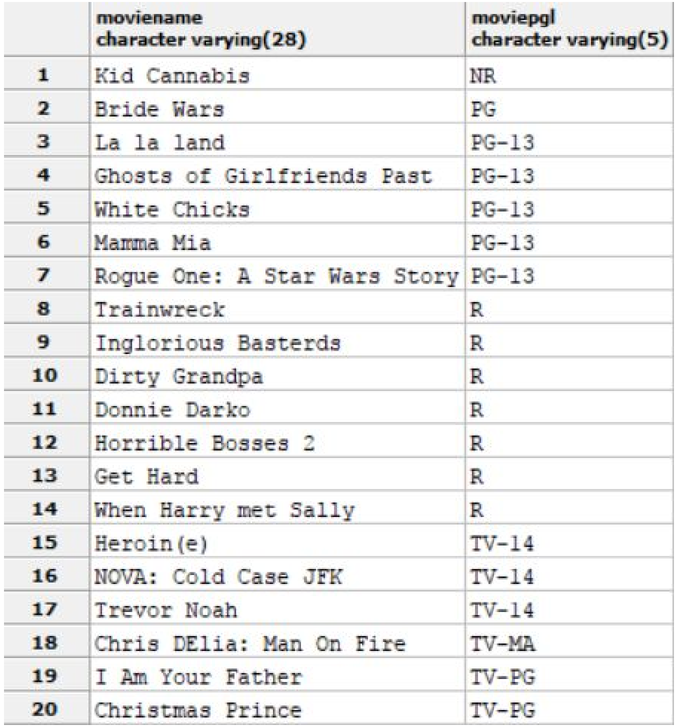
	moviename character varying(28)	moviepgl character varying(5)
1	Bride Wars	PG
2	Chris DElia: Man On Fire	TV-MA
3	Christmas Prince	TV-PG
4	Dirty Grandpa	R
5	Donnie Darko	R
6	Get Hard	R
7	Ghosts of Girlfriends Past	PG-13
8	Heroin(e)	TV-14
9	Horrible Bosses 2	R
10	I Am Your Father	TV-PG
11	Inglorious Basterds	R
12	Kid Cannabis	NR
13	La la land	PG-13
14	Mamma Mia	PG-13
15	NOVA: Cold Case JFK	TV-14
16	Rogue One: A Star Wars Story	PG-13
17	Saving Capitalism	TV-PG
18	Trainwreck	R
19	Trevor Noah	TV-14
20	When Harry met Sally	R

MOVIES RATING VIEW

This view gets all the parental guidelines ratings in groupings and their movie names.

```
CREATE OR REPLACE VIEW MOVIERATING
    AS SELECT DISTINCT
Movies.MoviePGL,Movies.MovieName
    FROM Movies;

 SELECT * FROM MOVIE
 ORDER BY MoviePGL ASC;
```

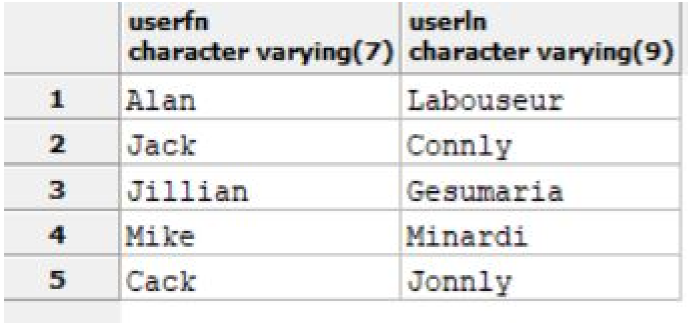


	moviename character varying(28)	moviepgl character varying(5)
1	Kid Cannabis	NR
2	Bride Wars	PG
3	La la land	PG-13
4	Ghosts of Girlfriends Past	PG-13
5	White Chicks	PG-13
6	Mamma Mia	PG-13
7	Rogue One: A Star Wars Story	PG-13
8	Trainwreck	R
9	Inglorious Basterds	R
10	Dirty Grandpa	R
11	Donnie Darko	R
12	Horrible Bosses 2	R
13	Get Hard	R
14	When Harry met Sally	R
15	Heroin(e)	TV-14
16	NOVA: Cold Case JFK	TV-14
17	Trevor Noah	TV-14
18	Chris DElia: Man On Fire	TV-MA
19	I Am Your Father	TV-PG
20	Christmas Prince	TV-PG

REPORTS

This report shows all the users who have recently watched documentaries.

```
SELECT Users.UserFN, Users.UserLN
 FROM Users
 WHERE EXISTS
          (SELECT *
           FROM RecentlyWatchedMovies, RecentlyWatchedTVShows
           WHERE EXISTS
                    (SELECT *
                     FROM Movies, TV_Shows
                     WHERE EXISTS
                              (SELECT *
                               FROM Genres
                               WHERE Genres.GenreName = 'Documentary'
                               AND Users.UserID = recentlywatchedtvshows.userid
                               AND Users.UserID = recentlywatchedmovies.userid
                               AND Genres.GenreID = tv_shows.GenreID
                               AND Genres.GenreID = movies.GenreiD)));
```

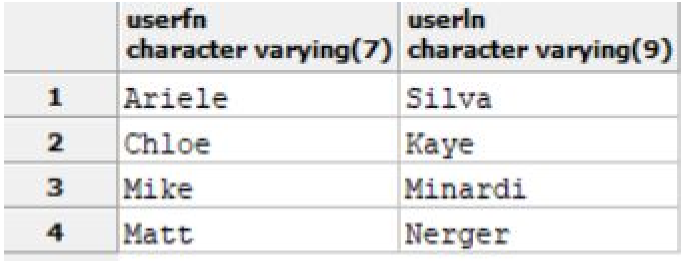


	userfn character varying(7)	userln character varying(9)
1	Alan	Labouseur
2	Jack	Connly
3	Jillian	Gesumaria
4	Mike	Minardi
5	Cack	Jonnly

REPORTS CONT.

This report shows all the users who have recently watched the newest season of Stranger Things.

```
SELECT Users.UserFN, Users.UserLN
FROM Users
WHERE EXISTS
        (SELECT *
         FROM RecentlyWatchedtTVShows
         WHERE EXISTS
                (SELECT *
                 FROM TV_Shows
                 WHERE TV_Shows.TV_Name = 'Stranger Things'
                 AND Users.UserID = RecentlyWatchedTVShows.UserID
                 AND RecentlyWatchedTVShows.TVID =
TV_Shows.TVID));
```

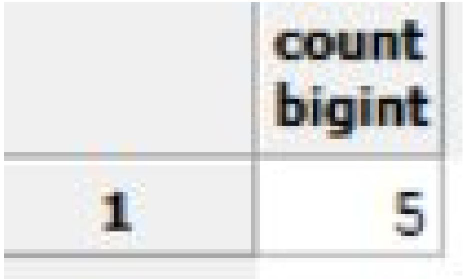


	userfn character varying(7)	userln character varying(9)
1	Ariele	Silva
2	Chloe	Kaye
3	Mike	Minardi
4	Matt	Nerger

REPORTS CONT.

This report shows the number of recently watched TV shows that user’s with the last name “Labouseur” has watched.

```
SELECT COUNT(RecentlyWatchedTVShows.UserID)
FROM RecentlyWatchedTVShows
WHERE EXISTS
        (SELECT *
         FROM Users
         WHERE Users.UserLN = 'Labouseur'
         AND Users.UserID = RecentlyWatchedTVShows.UserID);
```



	count bigint
1	5

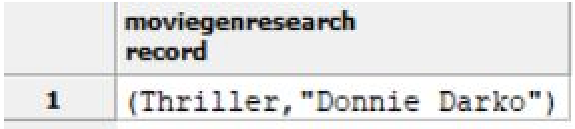
STORED PROCEDURES MOVIEGENRESEARCH

This function is used to search for movies based on their genre. By typing in a genre and running the function it will display all the movies that follow that genre in the Netflix Database.

```
DROP FUNCTION IF EXISTS MovieGenreSearch(GenreName VARCHAR);
CREATE OR REPLACE FUNCTION MovieGenreSearch(VARCHAR) returns TABLE(GenreName VARCHAR,
MovieName VARCHAR) as
$$
DECLARE
    FindGenre VARCHAR := $1;
BEGIN
    RETURN QUERY
    SELECT Genres.GenreName, Movies.MovieName
    FROM Genres, Movies
    WHERE Genres.GenreID = Movies.GenreID
    AND Genres.GenreName = FindGenre;
END;
$$ language plpgsql;
```

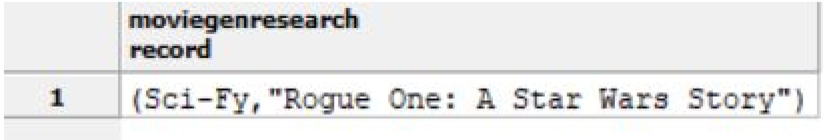
STORED PROCEDURES MOVIEGENRESEARCH CONT.

```
SELECT DISTINCT MovieGenreSearch('Thriller');
```



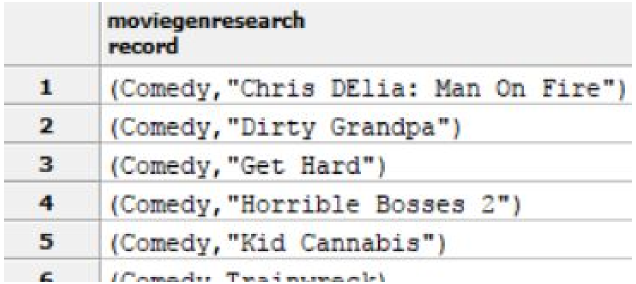
	moviegenresearch record
1	(Thriller,"Donnie Darko")

```
SELECT DISTINCT MovieGenreSearch('Sci-Fy');
```



	moviegenresearch record
1	(Sci-Fy,"Rogue One: A Star Wars Story")

```
SELECT DISTINCT MovieGenreSearch('Comedy');
```



	moviegenresearch record
1	(Comedy,"Chris DElia: Man On Fire")
2	(Comedy,"Dirty Grandpa")
3	(Comedy,"Get Hard")
4	(Comedy,"Horrible Bosses 2")
5	(Comedy,"Kid Cannabis")
6	(Comedy,Trainwreck)

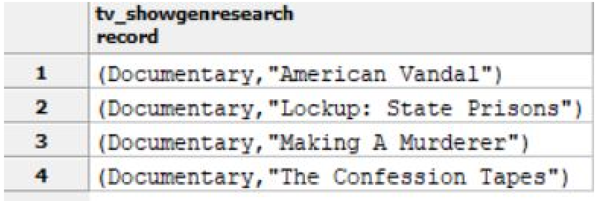
STORED PROCEDURES TV_SHOWGENRESEARCH

This function is used to search for TV shows based on their genre. By typing in a genre and running the function it will display all the TV shows that follow that genre in the Netflix Database.

```
DROP FUNCTION IF EXISTS TV_ShowGenreSearch(GenreName VARCHAR);
CREATE OR REPLACE FUNCTION TV_ShowGenreSearch(VARCHAR) returns TABLE(GenreName VARCHAR,
TV_Name VARCHAR) as
$$
DECLARE
   FindGenre VARCHAR := $1;
BEGIN
   RETURN QUERY
   SELECT Genres.GenreName, TV_Shows.TV_Name
   FROM Genres, TV_Shows
   WHERE Genres.GenreID = TV_Shows.GenreID
   AND Genres.GenreName = FindGenre;
END;
$$ language plpgsql;
```

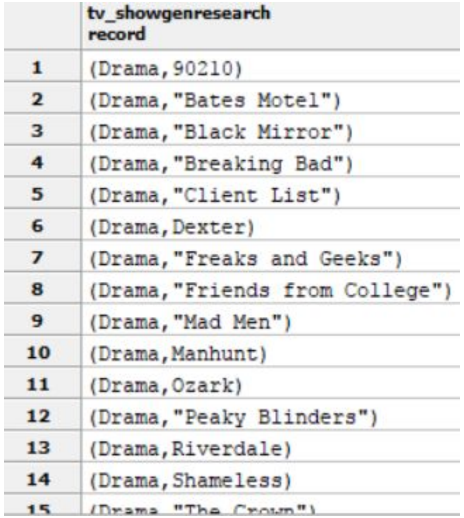
STORED PROCEDURES TV_SHOWGENRESEARCH CONT.

```
SELECT DISTINCT
TV_ShowGenreSearch('Documentary');
```



	tv_showgenresearch record
1	(Documentary,"American Vandal")
2	(Documentary,"Lockup: State Prisons")
3	(Documentary,"Making A Murderer")
4	(Documentary,"The Confession Tapes")

```
SELECT DISTINCT TV_ShowGenreSearch('Drama');
```



	tv_showgenresearch record
1	(Drama,90210)
2	(Drama,"Bates Motel")
3	(Drama,"Black Mirror")
4	(Drama,"Breaking Bad")
5	(Drama,"Client List")
6	(Drama,Dexter)
7	(Drama,"Freaks and Geeks")
8	(Drama,"Friends from College")
9	(Drama,"Mad Men")
10	(Drama,Manhunt)
11	(Drama,Ozark)
12	(Drama,"Peaky Blinders")
13	(Drama,Riverdale)
14	(Drama,Shameless)
15	(Drama,"The Crown")

SECURITY / ROLES

The admin role allows the Netflix administrator to add, delete, alter, etc. all the tables within the database.

```
CREATE ROLE ADMIN;
GRANT ALL ON ALL TABLES IN SCHEMA
PUBLIC TO ADMIN WITH GRANT OPTION;
```

The users role allows the users to select and view the different types of genres, and all the TV shows and movies within the Netflix database.

```
CREATE ROLE USERS;
GRANT SELECT ON Genres, TV_Shows,
MoviesTO USERS;
```

IMPLEMENTATION NOTES

While creating this database, I took the liberty to ask some friends for their Netflix data to get the most diverse and accurate data as possible. I also added some shows to the list, although this is not a complete Netflix database with every single show, movie, and genre; this is just a small portion. For the stored procedures that search by genre, be sure to search with correct spelling, for it will not take grammatical errors. Each TV show and Movie only has one genre, although they may fall into many different categories, I took the most obvious and dominant genre for the purpose of this project.

TRIGGERS / KNOWN PROBLEMS / FUTURE ENHANCEMENTS

Triggers are a special kind of stored procedure. They are the kind that automatically executes when an event occurs in the database server.

Some known problems are that multiple shows can have many different types of genres, and there needs to be an easier way to categorize each show's multiple genres without redundancy.

As a future enhancement, some of the stored procedures can be later altered to work alongside triggers. For example, once a new season is added to a show on Netflix it should trigger the DateChecked field to update the date. This will ensure that we know when the last time the show was updated, etc.