

FLIXWATCHER-FW

Barış Mert Bener - 26538

Murat Demiraslan - 27869

Görkem Yar - 27970

Furkan Kerim Yıldırım - 28138

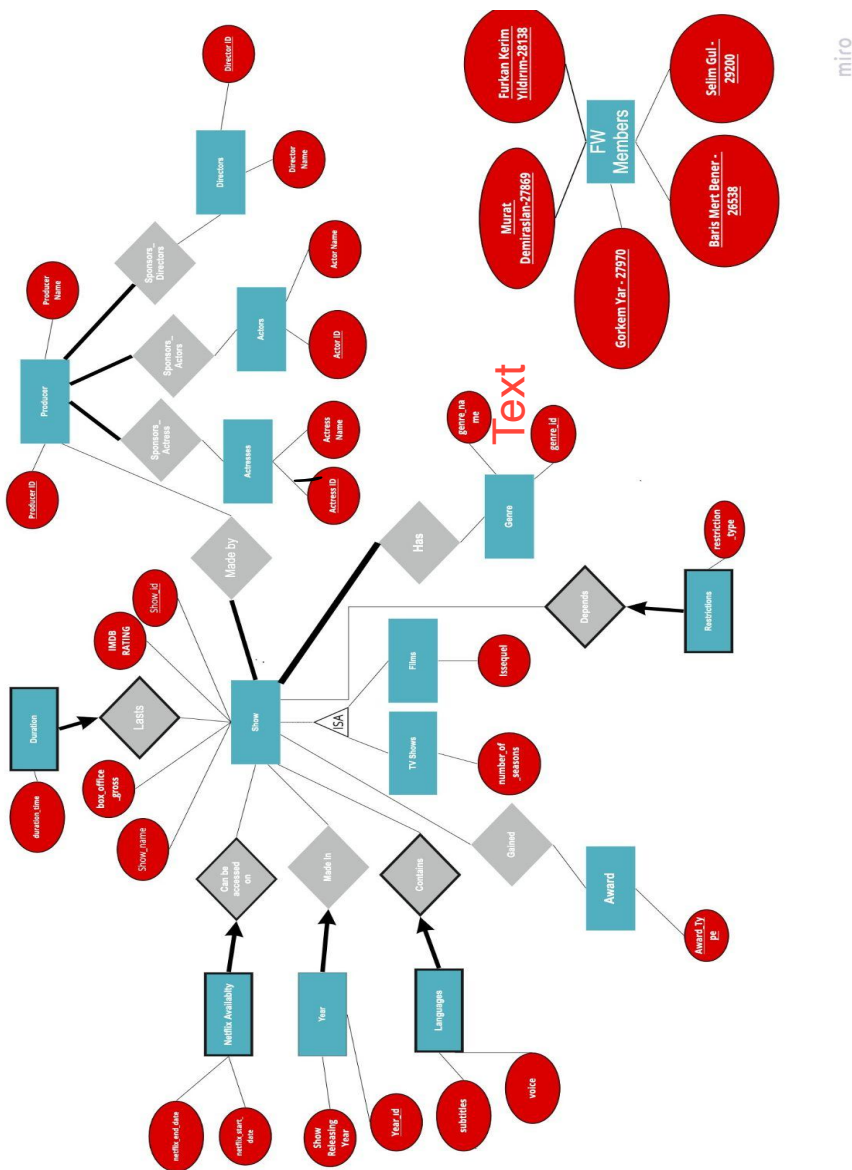
Selim Gül - 29200

In this project, we are aiming to create a database application that warns the Netflix user about the shows that are going to be deleted from the Netflix film database. To create our database, we are going to use the Netflix film database along with the Imdb's and Movie Db's databases. Firstly, we will obtain the information of some attributes in the entities from Netflix. After that, we will use the movie name, which is a unique one, to acquire the missing information from the other databases. We are planning to employ the MySQL as the database management system to create our database. As stated in the project description, we are going to use $(x+1)*2$ entities, which is 14. These 14 entities will help us to classify the distinctive attributes of the movies. The 14 entities are shows, movies, TV shows, year, Netflix availability, duration, producer, actors, actresses, directors, genre, restrictions, language, award.

First of all, shows as an entity is going to have 4 attributes which are show name that holds the name of the show and show id that will have a unique id of the movies to avoid using the same show names for shows having the same names, and IMDB rating, box-office gross which is holding total profit. Secondly, in order to identify the type of this show, two entities will be employed by using an ISA relationship. One of these two entities is TV series that have one attribute which is the number of seasons. Another entity is movies that have an attribute that examines whether the movie is a sequel or not. Apart from the ISA hierarchy, when it comes to duration of the movies, the weak entity of duration will be responsible for the total duration of the shows. Furthermore, the languages of the shows will be held by the language entity which has two attributes. One of them is the voice that determines if the movie is filmed originally in a language, and the other one is subtitles which holds the subtitle languages. Moreover, we are planning to include the adding and excluding date for shows in Netflix so as to be able to notify the user when the show will be excluded from the Netflix, and in order to be able to make this desire come true, we will have to use an entity named Netflix and two attributes of it named as start and end. Also, for the years of the movies, the year entity will be used and with this entity, there will be two attributes that year_id and the releasing year. In addition, there are four entities that are for the production part of the show. These entities are producer, actors, actresses, directors entities. The producer entity, which has the producer name attribute, sets three different sponsor relationships with the actors, actresses, and the directors entities. Actor, actress, and directors entities have two attributes, which ID and name of the particular entity. Also there is a restriction entity, which is a weak entity, and has an attribute named restriction

type. In terms of determining the success of the shows, the award entity will be used to hold the rewards such as Oscar, Emmy, etc. the movie has won since it has been first shown in theatres.

In the end, the Flixwatcher database is going to be linked to a web-based interface. This web-based application will be generated via HTML to compose web forms, and PHP scripts will be used for the processing side of the server. We hoped that our database system would be adequate to solve the queries.



```
CREATE TABLE Shows(  
    Shows_ID INTEGER NOT NULL,  
    Shows_Name TEXT,  
    Box_Office_Gross DOUBLE,  
    IMDB_Rating DOUBLE,  
    Primary Key (Shows_ID)  
);
```

```
CREATE TABLE Directors(  
    Director_ID INTEGER NOT NULL,  
    Director_Name TEXT,  
    Primary Key (Director_ID)  
);
```

```
CREATE TABLE Producers(  
    Producer_ID INTEGER NOT NULL,  
    Producer_Name TEXT,  
    Primary Key (Producer_ID)  
);
```

```
CREATE TABLE Year_Made_in(  
    Releasing_Year INTEGER,  
    Year_ID INTEGER,  
    Shows_ID INTEGER NOT NULL,  
    PRIMARY KEY (Year_ID),  
    FOREIGN KEY (Shows_ID) REFERENCES Shows(Shows_ID) ON DELETE NO  
ACTION  
);
```

```
CREATE TABLE Actors(  
    Actor_Id INTEGER NOT NULL,  
    Actor_Name TEXT,  
    PRIMARY KEY (Actor_id)  
);
```

```
CREATE TABLE Actresses(  
    Actress_ID INTEGER NOT NULL,  
    Actress_Name TEXT,  
    PRIMARY KEY (Actress_ID)  
);
```

```
CREATE TABLE Genre(  
    Genre_Name TEXT,  
    Genre_ID INTEGER NOT NULL,
```

```

        PRIMARY KEY (Genre_ID)
    );

CREATE TABLE Award(
    Award_Type VARCHAR(50),
    PRIMARY KEY (Award_Type)
);

CREATE TABLE Films(
    Is_sequal BINARY,
    Shows_ID INTEGER NOT NULL REFERENCES Shows(Shows_ID),
    PRIMARY KEY (Shows_ID)
);

CREATE TABLE TV_Shows(
    Seasons_num INTEGER,
    Shows_ID INTEGER NOT NULL REFERENCES Shows(Shows_ID),
    PRIMARY KEY (Shows_ID)
);

CREATE TABLE Netflix_Availability(
    Start_Date DATE,
    End_Date DATE,
    Shows_ID INTEGER NOT NULL,
    PRIMARY KEY (Shows_ID),
    FOREIGN KEY (Shows_ID) REFERENCES Shows(Shows_ID) ON DELETE CASCADE
);

CREATE TABLE Languages(
    Subtitle TEXT,
    Voice TEXT,
    Shows_ID INTEGER NOT NULL,
    PRIMARY KEY (Shows_ID),
    FOREIGN KEY (Shows_ID) REFERENCES Shows(Shows_ID) ON DELETE CASCADE
);

CREATE TABLE Restrictions(
    Restriction_type TEXT,
    Shows_ID INTEGER NOT NULL,
    PRIMARY KEY (Shows_ID),
    FOREIGN KEY (Shows_ID) REFERENCES Shows(Shows_ID) ON DELETE CASCADE
);

```

```
CREATE TABLE Duration(  
    Duration_time INTEGER,  
    Shows_ID INTEGER NOT NULL,  
    PRIMARY KEY (Shows_ID),  
    FOREIGN KEY (Shows_ID) REFERENCES Shows(Shows_ID) ON DELETE  
CASCADE);
```

```
CREATE TABLE Sponsors_Actress(  
    Producer_ID INTEGER NOT NULL,  
    Actress_ID INTEGER,  
    Primary Key(Producer_ID),  
    FOREIGN KEY (Producer_ID) REFERENCES Producers(Producer_ID) ON DELETE  
CASCADE,  
    FOREIGN KEY (Actress_ID) REFERENCES Actresses(Actress_ID) ON DELETE  
CASCADE);
```

```
CREATE TABLE Sponsors_Actor(  
    Producer_ID INTEGER NOT NULL,  
    Actor_ID INTEGER,  
    Primary Key(Producer_ID),  
    FOREIGN KEY (Producer_ID) REFERENCES Producers(Producer_ID) ON DELETE  
CASCADE,  
    FOREIGN KEY (Actor_ID) REFERENCES Actors(Actor_ID) ON DELETE CASCADE  
);
```

```
CREATE TABLE Sponsors_Director(  
    Producer_ID INTEGER NOT NULL,  
    Director_ID INTEGER,  
    Primary Key(Producer_ID),  
    FOREIGN KEY (Producer_ID) REFERENCES Producers(Producer_ID) ON DELETE  
CASCADE,  
    FOREIGN KEY (Director_ID) REFERENCES Directors(Director_ID) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE Gained(  
    Shows_ID INTEGER,  
    Award_Type VARCHAR(500),  
    Primary Key(Shows_ID, Award_type),  
    FOREIGN KEY (Shows_ID) REFERENCES Shows(Shows_ID) ON DELETE CASCADE,  
    FOREIGN KEY (Award_type) REFERENCES Award(Award_type) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE Has(  
    Shows_ID INTEGER NOT NULL,  
    Genre_ID INTEGER,  
    PRIMARY KEY(Shows_ID),  
    FOREIGN KEY (Shows_ID) REFERENCES Shows(Shows_ID) ON DELETE CASCADE,  
    FOREIGN KEY (Genre_ID) REFERENCES Genre(Genre_ID) ON DELETE CASCADE);
```

```
CREATE TABLE Made_By(  
    Shows_ID INTEGER NOT NULL,  
    Producer_ID INTEGER,  
    Primary Key (Shows_ID),  
    FOREIGN KEY (Shows_ID) REFERENCES Shows(Shows_ID) ON DELETE CASCADE,  
    FOREIGN KEY (Producer_ID) REFERENCES Producers(Producer_ID) ON DELETE  
CASCADE);
```