Part 1: Designing a Database

The dataset we obtained is a subset of data available from IMDb which is located at: https://www.imdb.com/interfaces

The dataset is a comprehensive collection of all the movies/TV shows (among other things) found on IMDb along with information regarding the actors and other cast members that contributed to the production of the motion picture and the ratings. The collection is extensive with data dating way before the 1800s. We have trimmed the data such that the actors with a date of birth prior to 1935 have been excluded as well as titles that were released prior to 1980. We have also trimmed the TV shows to only include a select few that are our favorites to reduce the size of the episode dataset.

The ER diagram depicts the project entity at the core of the diagram. This is the general superclass for all motion pictures. The Movies and Series subclasses refer to movies and TV shows, respectively. All projects will have multiple cast members and actors, the same person may be involved in multiple projects. A project may have multiple applicable genres and a person may have multiple applicable professions, hence the multivalued attribute for each. Each project has an associated average rating along with the number of votes that contributed to that rating. The project specialization is disjoint as a project cannot simultaneously be a movie and a TV show.

Cardinality constraints:

One-to-one(1:1): Each rating is associated with one project, one project only has one rating

One-to-Many(1:N): One series can have many episodes, many episodes can only belong to one series.

Many-to-many (M:N): A person can be a cast member involved with many projects, a project will hire multiple cast members. A person can be a creator/crew member involved with many projects, a project will hire multiple crew members.

Participation constraints:

Total Participation:

1) A project must fall under either the movie or TV show category as that is the focus of this database, other motion pictures have been excluded.
2) A TV series must have an episode
3) An episode must be associated with a TV series
4) Each project must hire cast members
5) All ratings are associated with a project

Partial participation:

1) A project may not necessarily have a rating or it may still be in production and has not yet been released.
2) Not all people work on a project

Translating ER diagram to relational model:

1) Mapped the strong entities: Project and Person. Created a relation for each, all attributes were included. Primary key for Project and Person was chosen to be tconst, nconst, respectively.
2) Created a relation for the Rating weak entity. The foreign key for the Rating entity is the tconst referencing the Project relation.
3) Mapped 1:1 relationships: Project and Rating. We included the primary key (tconst) from the Project relation as the foreign key in Rating which has total participation.
4) Mapped 1:N: Series and Episodes. We included the primary key (tconst) referencing the Project relation as the foreign key in the Episode relation.
5) Mapped N:M. A cast relation was created that includes the primary key nconst from the Person relation and the primary key tconst from the Project relation as foreign keys in Cast.
6) Mapping multivalued attributes:
    - A Profession relation was created. The primary key of Person, nconst, was included as a foreign key in the Profession relation
    - A Genre relation was created. The primary key of Project, tconst, was included as a foreign key in the Genre relation.
7) Mapping specialization: Project, Movies, Series was mapped using the multiple relations-superclass and subclasses method. A Project, Movie and Series relation were created.

Normalization

- 1NF has been satisfied as the multivalued attributes, Profession and Genre were broken up into separate tables to ensure atomic values are present in each column of the Project and Person relations
- 2NF has been satisfied as the only two relations with multiple primary key attributes is the creator/crew and Actor relation. Both non key attributes are dependent on both tconst and nconst and therefore satisfy 2NF.
- 3NF is satisfied, no relation has non primary attributes that are dependent on one another.