ERD WRITE-UP

PROJECT 5 - DESIGN

BRENNAN W. FIECK

CSCI 403A
DATABASE MANAGEMENT
COLORADO SCHOOL OF MINES

PROFESSOR:

DOCTOR CHRISTOPHER PAINTER-WAKEFIELD

MARCH 1ST, 2017

Contents

1

OVERVIEW

The full Entity Relation Diagram (ERD) is given in Figure 1.

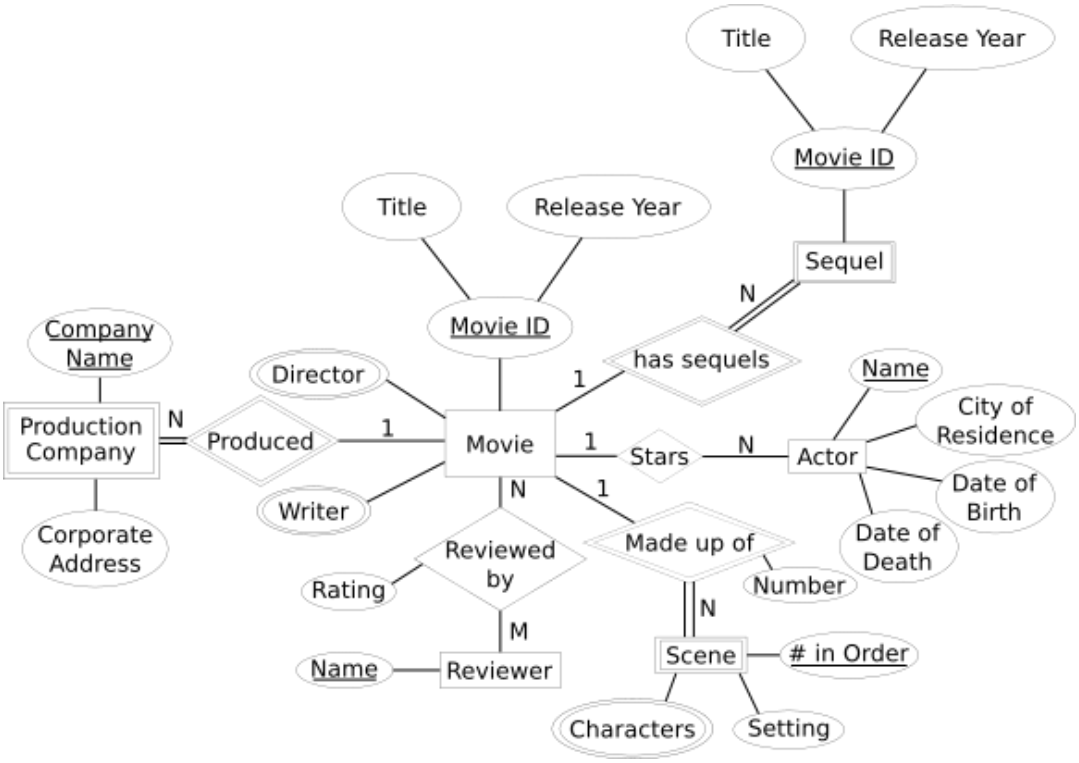

Figure 1: Full ERD for Movie Database

At the center of the diagram is the 'Movie' entity , which was chosen as a "main" entity for obvious reasons. The unique identifier or key was chosen as a composite of a movie's title and release year, because remakes occur but not fast enough to occur in the same year as the original. Directors and writers were chosen to be multivalued attributes because they presumably only have meaning in the context of a movie, and their job is more or less fully described by the name. Also, a movie may have one or more writer(s) or director(s). Reviewers, actors, sequels, and production companies were chosen to be entities, for varying reasons. Sequels because a sequel necessarily describes a movie itself, and so is something a user would conceivably want information about. Actors were chosen to be entities for similar reasons, whereas Reviewers were chosen to be entities to preserve a specific N-to-M relationship and maintain a specific rating for a specific review of a specific movie.
Production companies were chosen to be entities because they contain a set of data pertaining to a certain company in particular and

need to be uniquely identified. It's a weak entity because a production company produces movies, and therefore is useless without a definition of a movie. It's assumed that production company names are trademarked and therefore unique, and the corporate address is stored as an attribute.

The additional entity I chose to incorporate was a scene from a movie.

2

ACTOR

Figure 2 shows a partial ERD defining the 'Actor' entity.

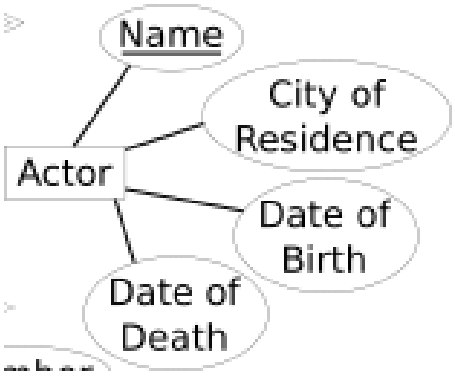

Figure 2: Partial ERD for 'Actor' Entity

This model makes the assumption that an actor's name uniquely identifies it, but other than that is not terribly complicated. It contains attributes describing the actor's city of residence, date of birth and date of death. The information about an actor's age could possibly be another attribute, but that would be extraneous because that information is fully contained by the date of birth and date of death. Personally, I wouldn't store that information at all, especially since it will become incorrect once a year and have to be updated; anything or anyone that wants to query the database to find that information can perform calculations at that time, which will ensure consistent correctness (assuming the database is updated properly on the event of an actor's death). The 'Actor' entity is not a weak entity, so that searches over actors can be made and thus the information about a list of movies acted in by a certain actor can be extracted.

3

SEQUEL

Figure 3 shows a partial ERD defining the 'Sequel' entity. It's a weak entity, because a sequel cannot exist if there's no original movie for it to be a sequel to.

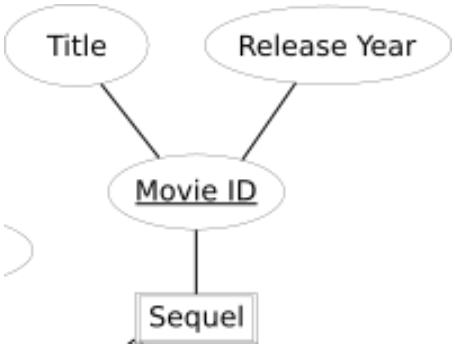

Figure 3: Partial ERD for 'Sequel' Entity

A sequel is fully defined by its title and release year, as well as the "owning" movie that it is a sequel to. A movie can have any number of sequels, each its own movie, so the key that identifies a sequel could also identify a movie with its own ERD from Figure 1.

4

SCENE

Figure 4 shows the definition for my additional entity, the 'Scene' object. Every movie is made up of a series of scenes, and each scene is uniquely identified by the order in which it occurs in the movie (two scenes cannot occur simultaneously, after all).

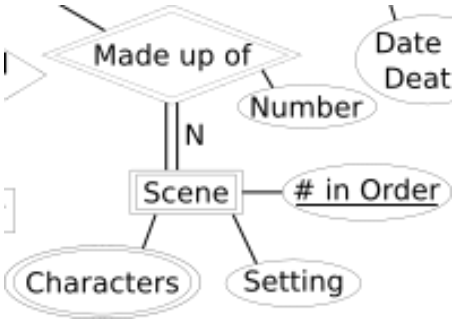

Figure 4: Partial ERD for 'Scene' Entity

A scene is a weak entity because it's just a part of a movie. It has an associated setting, and any number of participating characters. Conceivably, there is much more information that could be contained in this entity, but that would get exhaustive and frankly, difficult to read. The relation is associated with a number so that a query wouldn't have to count scenes to find the number of scenes in a movie. This information is extraneous, but unlike in the 'Actor' entity, which doesn't directly represent age because finding it is a constant-time operation from the information that *is* stored, finding the number of scenes without storing it directly is an $O[n]$ operation.