Movie Design

This activity will be very similar to the ward member activity but this time you're going to do one on our own. You'll look at the data in this file and you're going to decide which are entities and what attributes go with that entity. You'll also have to decide if it's really an entity or just an attribute of another entity. At this point we're going to assume that we have asked all the in-depth questions and gone through the whole statement of work process with our users or owners of the system, before we start designing. You are going to have to make some assumptions. For example, I'll bring up some assumptions you can make with studio later but I want you to think through it and your design may depend on some of those assumptions, and that's fine, so some people's designs might be different from other peoples. But be ready to talk about what those assumptions were. At this point try the design on your own and then wait to watch the next video until you're done.

_____

This is a an ERD that I came up with, but it doesn't mean that yours is wrong. I will show you how I thought through it. I, of course, had movies as an entity. I decided that title and release_date were attributes that fit with this entity. I knew from first normal form definition that genre, features and actors were separate entities because there were multiple values possible of these for each movie.

But what about studio, media type and rating. Let's look at each of those.

Here's where you would ask the owners of the system exactly what output they would need from the system to know exactly what you'd need to design into the system. We need to make some assumption here since we can't ask. Of course, studio could be a separate entity. It is a real-world object that we can model. But in our case, we only have one value for each movie. If we find that there will only ever be one studio per movie, and we won't ever need to have any more information about that studio like it's address or location, then it might be OK to leave it as an attribute of movie. But if you decide you could possibly have more than one studio per movie or that in the future you might want to include more information about studio or they might also want to include studios that aren't related to a movie yet in their database. Then it might be wise to make studio its own entity. I am going to assume those needing the database might need studio as a separate entity, but I'm also going to assume that there is only one studio per movie though, as is shown on our data. I'll assume a one to many relationship with the many part being on movies, studios can make many movies but each movie will have only one studio. Here is what the tables would look like with data. The 'many' table having the foreign keys that can repeat.

Let's look at media type. There is only one media type listed per movie, but could each of those movies be offered in other media types? Yes. So I'll make the assumption that each movie could be a DVDs and blu-ray and digital streaming, etc. And each media type, for example DVD can be the media for a lot of different movies. (for example, many movies are available in DVD) So I'm going to make a media_type table with a many to many relationship with movies. Let's look at all the relationship of actors, genre, and features because they are all similar.

An actors can be in many movies and a movie can have many actors in it.
A movie can belong to many different genres and a genre can be used in different movies.
A movie can have different features and each feature can be in many different movies.

Now where would price fit? Let's assume the price is the manufactures release price (the price printed on the media, the manufactured list price. If we put price in the movies table, we are saying that movie is always that price. That doesn't make sense because it depends on if it's a price in the theater, on DVD, on blu-ray, download, VHS etc. If we put price in the media_type table we are saying all DVDs are 19.95 and all blu-rays are 29.95 no matter what movie they are. Is this true? No, so where would price go? Well it depends on the movie and the what media type it is. So it belongs in the linking table with the composite primary key. This is an example of second normal form, which states all non-key attributes must rely on both parts of the primary key to belong to that entity. The non-key attribute is price, and does it rely on not only what media type it is but also what movie it is as well? Yes. So it does belong the and passes 2nd normal form. Here's what that would look like with data. Empire Strikes back is offered in DVD and blu ray and the price depends on the movie and the media type. Empire Strikes Back DVD is 19.95 and Empire Strikes Back blu ray is 29.99.

What about rating? Can't a lot of movies have the same rating and then each movie has one rating. Would this be a look up table? With the ratings listed in its own table? I would argue no, unless I needed more information about that rating like the description of what G or PG actually means, I would not make it its own entity. If I leave it as an attribute of movie and then later decide I want to describe what each one means for example G is for General Audience and PG is for Parental Guidance etc, why couldn't I just add that attribute to the movie entity as well. So it looks something like this.

This would not work because now we are not in 3rd normal form which states that each non-key attribute has to relate to only the primary key. As we go through all the attributes that are not the primary key let's see how this works with 3rd normal form. Does the title of the movie depend on that primary key? Yes that primary key will only ever represent that movie. Will the release date always go with that movie and primary key? Yes. Will the rating always go with that primary key for that movie? Yes. Now look at the rating description, does that rely on the movie primary key or the rating itself? The description of General Audience depends on the rating G, not the primary key of the individual movie. So we know that does not belong in the entity of movie to be in 3rd normal form. We would need to take rating and description out and put them in their own entity and relate them to each movie or in our case I will just take the description out because we don't need it in our system. Now we are in 3rd normal form.

So there is another example of taking data and separating it out into entities and making sure each entity has the proper attributes.