Critique of the current system

## Relationships

To begin, the system in its current form can’t be converted into a relational database due to the duplication of potential foreign key fields (KNOWNFORTITLES 1-4) and also multiple instances of the same TITLEID in different records. This is the case as in the Titles table, the title id isn’t the primary key of the title and therefore would need to be a composite primary key. This causes issues as the composite data can’t be paired with the title id (as found in the dump). This ultimately means that while it may not look like it, there is no relationship between the data as there is no one field that uniquely identifies the records.

## Associations

While the existing system has no relationships between the entities, there are associations between the KNOWNFORTITLES, there is an association between that field and the TITLEID and also an association between the NAMEID and the names in the title table. It’s a good idea to try and preserve uniquely identifiable data for individual records as eventually, 1 NAMEID will identify 1 person and 1 TITLEID will identify 1 title.

## Duplicate attributes

For the names dump, there are multiple profession fields, e.g. PROFESSION1, PROFESSION2. This should be represented as a separate entity and should be a one to many relationships, with one person being able to have many professions. There is also multiple known for fields, e.g. KNOWNFORTITLES1, KNOWNFORTITLES2. This shouldn’t be included in the final system as people and films will have relationships between them.

There are also multiple director fields, e.g. DIRECTOR1, DIRECTOR2. This should be represented as a separate entity, with the TITLE\_ID attribute as the foreign key. Should be a one to many relationships, with one title entity being able to have multiple directors. The same issue exists with writers, WRITER1, WRITER2. This should be represented as a separate entity, with the TITLE\_ID attribute as the foreign key. This too should be a one-to-many relationship.

There are multiple writer fields, e.g. Multiple genre fields, e.g. GENRE1, GENRE2. This should be represented as a separate entity, with the TITLE\_ID attribute as the foreign key. This should be a one to many relationships, one title entity and multiple genres. This is an issue though as many titles can cover many genres, therefore, we will need to include a weak entity between the 2.

Lastly, there is a duplication of PROFESSION fields which limits the number of roles one person can have. Ideally, we would have a separate entity for this or replace this with the individual entity and branching off this for entities like WRITER and DIRECTOR.

## Redundancy

As well as duplicate attributes, there are also instances where the data itself is repeated in the touples. For example, the duplication of most titles as they have different principals. This increases the risk of data duplication and the risk of data inconsistency. It also creates issues when you want to Update, Delete or Insert as you’re not able to update all records simultaneously, you have to go through each record and update it, you also can’t delete records as other data is dependent on the existing data. If you had a relational model when you create data it relies on other data to be present and eliminates the risk of these anomalies so long as the data is in 3rd normal form or higher.