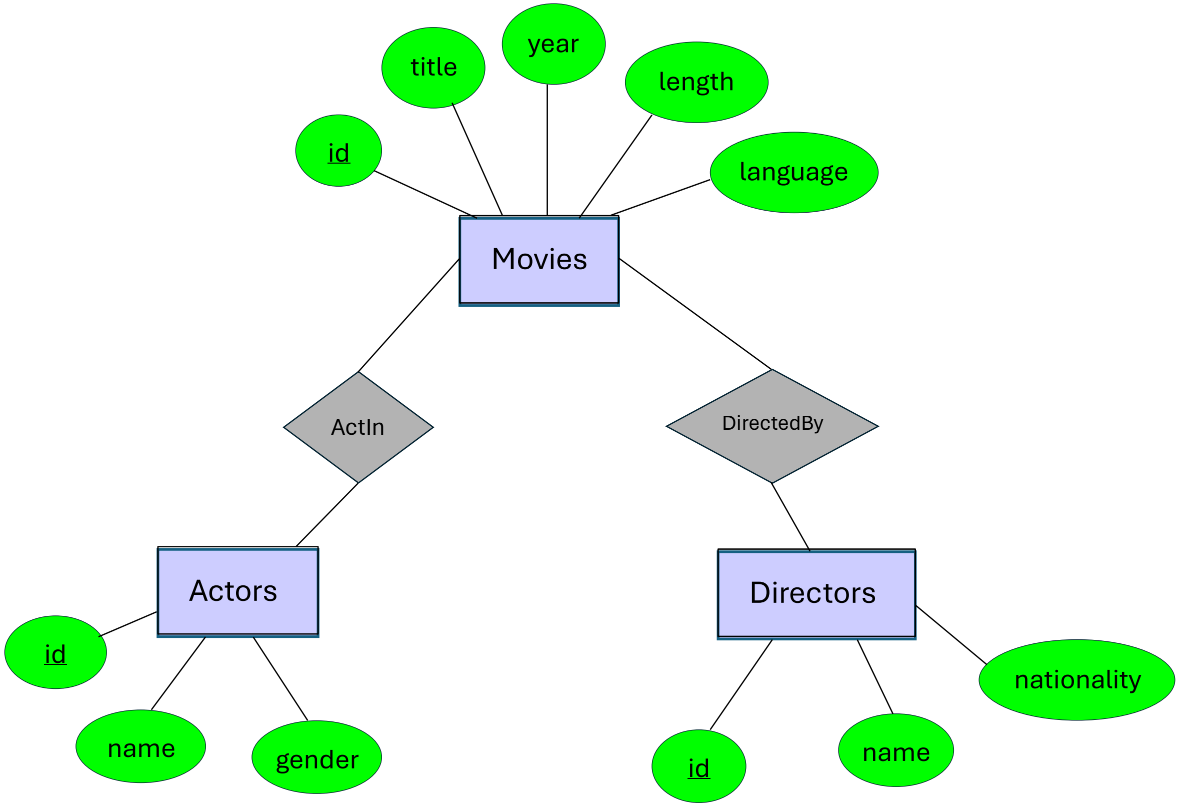
I designed this diagram based on the understanding that a movie only has one main director. In some cases a movie can be directed by multiple directors. For example in the 2006 movie "Paris, je t'aime" is directed by 10 directors. Thus, the relation 'DirectedBy' of Movies and Director is **many-to-many**. This is shown in my diagram below.

We also won't assume that a movie must have at least one director. There are cases that in a database the director is missing (null). Therefore I chose not to apply referential integrity constraint here.

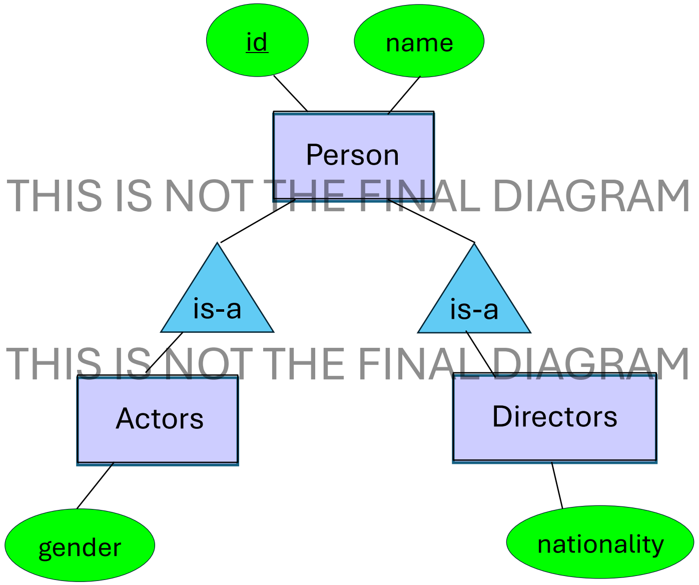
The primary keys of Movies, Actors and Directors are underlined.



CONTUNUE NEXT PAGE

It is also worth noting that Actors and Directors can be the subclasses of Person. In this case we would have to build another class Person with attributes id and name. There will be 2 ‘is-a’ relationship(shown below).

This design is gives more flexibility for expanding the database. If in the future, more common attributes are being added to Actors or Directors like years in the industry or awards winned, having a Person class will makes it easier and readable. However in this homework I will stick to the above version.


id
name
Person
is-a
is-a
Actors
Directors
gender
nationality