Important tables:

Movies

Has its own primary key

Has FK’s for genre, not really useful.

Users

Has its own primary key

Related to watch history and subscriptions.

Subscriptions

Has its own primary key

Many to one with Subscription plans and Users

Subscription Plans

Has its own primary key

Is separated out to ensure 3rd normal form.  
Watch\_history

Has its own primary key

One to many relationship with Users and Movies.  
Ratings

Has a joint identity FK/PK with both reviewers and movies.

Each rating is linked to one reviewer and one movie.

Reviewers:

Has its own primary key

Since subscriptions are a per user thing, I made subscription plans their own thing, which each subscription would refer to. – To allow subscription plans to change movie permission, I’ve created a linking table – In cases where there is no link entry present, it’s up to the implementing programmer to decide a sane default.

Watch history gets it’s own primary key instead of having just Fks, because a user can watch the same movie on multiple dates.

Genres is it’s own little FK thing.  
  
Reviewers act as a FK for ratings – An identity key at that, since I’m assuming the streaming platform does aggregates like “Minecraft Movie :Rotten Tomatos: 2.1”

Otherwise, it is all pretty simple.