

When designing the database schema for our final project, the YATV.app, we decided to create the following entities: User, App, Platform, Video, and Show. With these we created the following relationships: Likes (1 user likes 0 or many videos), List (1 user has a list with 1 or many videos they wish to watch), Host (1 video must have a single host), Subscription (1 app has 0 or many users), App-Platform (Many apps, can be on many different platforms), and Season (many videos can be on many groups of shows).

Non-technical users of the app should understand that a Video entity refers to a single video on a specific platform, this video can be part of a season as an episode. It is worth noting that a video or an episode on one platform may be exactly the same in terms of Name and description, between platforms, however, are described as different "videos" in the database. Users will have separate lists for shows or videos they have watched/like/want to watch and may subscribe to specific apps. Subscriptions are not specific to specific platforms.

In terms of our interpretation of the narrative, we made assumptions about the uniqueness of the names of Apps, Platforms, Shows, and user email addresses. We added an ID to Video as the primary key, as we decided that it was unreasonable to have uniqueness across all apps for video names. We also assumed that the same video or show cannot exist on multiple Apps. This would have slightly changed the design of show and season to decrease the amount of duplicate data.

In terms of restrictions, the only major restrictions we have is that a video cannot be present in multiple seasons and that a video must have a host. This restriction was given in the spec; however, I think it would be perfectly reasonable for a video to exist without an App connected to it. Say for example a user had a video in their list and it was removed briefly from a platform before appearing on another app, the user would still want their personal data about that video. For example, if they had watched it, or if they wanted to in the future.