

Name: Megha Samala

Table of Contents

Project Direction Overview.....	2
Use Cases and Fields.....	2-6
Structural Database Rules.....	6
Entity-relationship Diagram.....	7-8
Summary and Reflection.....	8-9

Project Direction Overview

I would like to create a web application designed to track a user's media consumption across several different mediums. The application, which I will call "TunedIn," will be a place where users can log the different media they're currently consuming, have consumed, or plan to consume. The media in question can be movies, TV shows, books, music (in album form), and video games. Additionally, the user will be able to track information about the vendor of the media and use this information to organize and revisit their media buys as well as give recommendations to others

I believe an application like this is important because an overall media tracking system does not appear to exist, leaving people to log their consumption on multiple different websites dedicated to only a specific medium (think Letterboxd or MyAnimeList), or create a Twitter thread or blog dedicated to tracking, or alternatively not track their watches/reads/listens at all. As someone who considers herself to be scatterbrained, I also often jump from media to media without finishing a series or completing a playthrough, and I forget what episode I left on or what chapter of a book I was on, and sometimes I completely forget that I was in the middle of watching or reading something. When people ask me what I've been watching or reading or playing, or what my favorite movie or show is, I often blank and fail to answer the question. I can safely presume that I am not the only one with this issue, and therefore an application dedicated to tracking the status of someone's varied media consumption is a convenient solution. I also believe that tracking media consumption encourages users to think more critically about what they are consuming and how they feel about it, rather than finishing a series or book or game and simply moving on to the next one.

TunedIn will offer a place for users to input information about a piece of media, the main focus being on a name, a rating (if completed/desired), and a status on the consumption. Additional information can be added based on relevance, such as author information for a book, or the URL for something consumed online so it can easily be provided to a friend when recommending the media to them, or a link to another existing database-esque website which may contain a blurb/critic ratings of the media (again, think Letterboxd or MyAnimeList or IMDB, etc), and more. Users will be able to log what chapter of a book they are on, or what episode of a show they are on, so that if they drop it and choose to resume it later, they can easily pick up from where they left off. Additionally, users can track when they started and finished a piece of media, to log past watches, and/or determine if it might be time to revisit something they previously consumed and enjoyed. Each kind of media has information that is specific to it, as well as information that can be applicable to multiple kinds of media, such as a genre and a review. Through my multiple project iterations, I will be able to refine my scope and determine what information is most necessary and relevant for the user to have logged, and create the most useful version of my database.

Use Cases and Fields

Use Case #1: Account Setup

Because TunedIn will be a web application, it will be necessary to register to keep track of each individual user's media.

Field	What it stores	Why it's needed
username	This stores a username	Users can share usernames to

	associated with each account	see what another user is consuming, and users can also have multiple accounts if they would prefer
first_name	This stores the user's first name.	This can be displayed on screen and shared with other users.
last_name	This stores the user's first name.	This can be displayed on screen and shared with other users.
account_created	This stores the date on which the user created the account.	Users may want to know how long they have been using the application to track their media consumption.

Use Case #2: Logging a Movie

A movie will be one of the media types that the application will encourage users to log, and has fields that are specific to it. The other media types will track similar information and will be able to be differentiated in the database using a flag.

Field	What it stores	Why it's needed
movie_name	This stores the name of the movie.	This is necessary for the user to know what movie they watched.
director	This stores the name of the director of the movie.	This is useful for the user if they want to track how much they like movies by a certain director or want to sort by a certain director.
production_company	This stores the name of the production company of the movie, such as A24 or Studio Ghibli.	This is useful for the user to track how much they like the works of a certain company or sort by them.
date_released	This stores the date that the movie was released.	This helps the user determine their viewing trends by date and also distinguish between potential movie remakes or remasters with different release dates.

Use Case #3: Logging a Genre

Since genres are applicable to any media type, users will be able to log them for any media consumed.

Field	What it stores	Why it's needed
category	This is a catch-all for the different media types where the category/large genres can be stored, such as things like fiction or nonfiction for books, documentary or short film for movies, RPG or shooter for video games, etc.	This allows for users to sort and search for certain pieces of media in a broad way.
main_genre	This stores the primary genre of a piece of media, which can differ in definition based on the user, but is generally a broader category such as horror, action, etc.	This allows for users to sort media by genre when organizing and also looking for previously logged media.
subgenre	This stores the subgenre of a piece of a media, which can be a specifier, like historical fantasy vs urban fantasy, etc.	This is optional but allows for more specific searching and sorting for the user.

Use Case #4 - Logging a Review

Since reviews are applicable to any media type, users will be able to log them for any media consumed.

Field	What it stores	Why it's needed
rating	This stores a number rating out of 10.	This is a typical method of rating not just media, but anything, and will let users reflect on what they enjoyed or disliked.
review	This stores a text review.	This lets the user give clarification on the number rating and offers more information about the media.
review_link	This stores an optional link that leads to an external blog review or a website like Letterbox or Goodreads that the user may have posted their review on.	This lets the user organize relevant info in one place by allowing them to keep track of different websites they may have used to review a piece of

		media.
--	--	--------

Use Case #5: Logging a Status

Each piece of media will have a consumption status that can be logged by the user.

Field	What it stores	Why it's needed
overall_status	This stores a general status like Planned, In Progress, Completed, Dropped, or On Hold.	This helps users keep track of all the media they want to watch, are watching, and have watched, as well as media they chose to not complete and media they plan to return to, which is handy for not forgetting their progress.
section	This stores a larger indicator of progress, such as a season of a TV show, a chapter for a book, a section/act of a video game, etc	This is helpful for giving context to the part of the media the user is on, or is an indicator that they finished it.
subsection	This stores a smaller indicator of progress, such as an episode of a TV show, a page of a book, a level of a video game, etc.	This is helpful for tracking the exact part of a piece of media that the user left off on so they can jump back in.
date_started	This stores the date when the user started consuming the media.	This is helpful for the user to know when they first began a piece of media and for tracking how long it is taking them to complete it.
date_finished	This stores the last date the user interacted with/finished the media.	This is helpful for tracking the amount of time the user has spent with the media and for tracking how long ago they interacted with the media to see if it is worth a revisit.

Use Case #6: Logging a Vendor

Media is offered by multiple vendors and these vendors can be frequented by users, and therefore are logged in the database.

Field	What it stores	Why it's needed
vendor_name	This stores the name of the vendor.	This is needed for the user to know what vendors they buyt from.
vendor_type	This stores the type of vendor, such as website, streaming service, or physical storefront.	This is useful for the user to know where they acquired a piece of media.
vendor_link	This stores a link to the vendor website, if available.	This can help the user track past visits to an online vendor and allow them to revisit it or recommend it.

Use Case #7: Vendor-Media Relationship

This connects the vendor to the piece of media and provides useful information about the purchase.

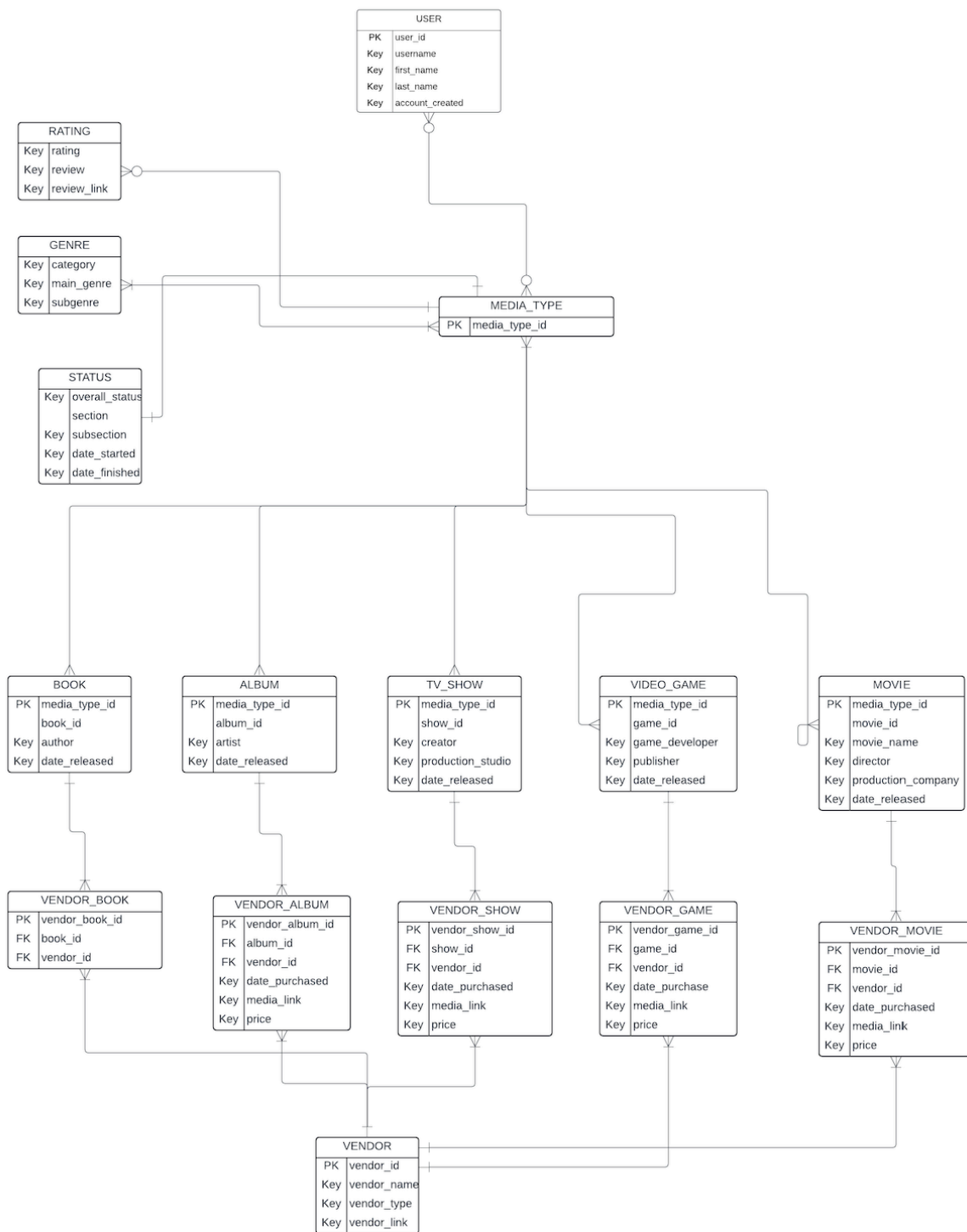
Field	What it stores	Why it's needed
vendor_name	This stores the name of the vendor and will reference the vendor table.	This lets users know where they bought or can buy a piece of media.
media_name	This stores the name of the media and will reference the media table.	This lets users know what media they bought/can buy from the vendor.
date_purchased	This stores the date of the purchase of the media from the vendor.	This can assist with tracking the consumption of media by time.
media_link	This stores a direct link to the media on the vendor site, if available/applicable	This can help the user revisit their purchase easier or recommend the purchase to a friend.
price	This stores the price of the media purchase, if applicable	This can help users budget and track their money spent on media and also play into their review of the media by determining if it was worth the price they paid for it.

Structural Database Rules

- 1) Each user may log zero to many types of media, and each media type may be logged by zero to many users.
- 2) Each media type may have zero to many reviews, and each review will apply to one media type.
- 3) Each media type will have one status, and each status will apply to one media type.
- 4) Each media type will have one to many genres, and each genre can apply to one to many media types.
- 5) A media type is a book, album, TV show, movie, or video game.
- 6) Each media type is purchased/streamed from a vendor, and each vendor can offer one to many media types.
- 7) Each media type can have one to many purchases, and each purchase can fall under one media type.

Conceptual Entity-Relationship Diagram

My entity-relationship diagram is pictured below and uses the Crow's Foot format to convey the structural database rules and relationships I delineated above. Because each media type entity is similar to the Movie use case, and each media type entity has a similar vendor relationship entity like Vendor_Movie, I chose not to create a separate use case for each but instead display them below.



Summary and Reflection

My database is for a web application called “TunedIn” in which users can track their media consumption across various mediums, as well as media purchases from various vendors. The application allows users to rate their media and track their progress in consuming it.

This week I was able to fix the structure of my database compared to my first iteration by taking some of the information I planned to log and separating it into different entities that would flesh out my database more. I am surprised to see that I do not have as many structural database rules as I thought I would considering the number of entities I have, and while I can see this being the case because my entities can be similar to one another, I wonder if I am perhaps missing some rules that may be necessary. Additionally, I am still wondering if I should flesh out my database structure further with more entities that I think of - for example, because video games and movies and tv shows have production companies and directors and producers, would it be wise to put that information in a separate entity that can then have a relationship with with those three media types? I would appreciate any and all feedback on things I could do to improve my project and clean it up/make it more coherent.