Adonias Daniel & David Newman
Professor Cano
CMSC 508
05.05.2024

Project Phase 3: AnimeCatalog

Problem Statement

AnimeCatalog is a database designed for users to explore, discover, and manage their anime-watching experience. The anime industry has recently increased in popularity all around the world, with a diverse selection of genres and animes growing by the day, there are countless gems to choose from. Due to this, the anime community has always had one issue, keeping track of all the wonderful animes we've watched! We often struggle to track new releases, follow our favorite series, or discover new anime that matches our preferences. With thousands of animes, theirs always another anime to watch and fall in love with, so it's always been hard to remember the spectacular ones we've watched in the past or planned to watch in the future. That's why AnimeCatalog aims to become the one-stop shop for all anime enthusiasts with features like searching for animes, creating watchlists, and finding new animes!

Search Function

The primary function of our database will be centered on the search function. With the search function users will have the ability to search for anime by title, enabling them to access detailed information. This includes the anime's review score, age rating, genre, summary, aired date, episodes, duration, type ('TV' or 'Movies'), studio, source type ('Manga' or 'Light novel'), language type: ('Sub' or 'Dub'), imageURL, and its streaming platform. This function caters to users who know what they are looking for and want specific information on certain anime titles.

List Management

A major function in our database will be list management functionality. This will allow users to create and manage a list of anime based on whether they’ve previously watched, are currently watching, or plan to watch in the future. After choosing an anime, you will be able to add them to either a watchlist or a completed list. These lists will be very beneficial in keeping track of all the animes you've watched, in order to recall them, or even share them with friends. The goal is to have users add the animes that catch their eye into a watchlist so they can keep track and remember to watch them in the future.

Random Anime Discovery

Another functionality is the random anime discovery. A "Pick For Me!" button will show users a random anime series or film with its corresponding information. Once the button is clicked and a random anime appears, users will be able to view the details of the anime and add them to a list, with the hope of allowing users to eventually stumble upon an anime that catches their attention. This feature aims to help users find new animes they otherwise would have never seen while simultaneously helping users find hidden gems. The random anime discovery will hopefully expand users' preferences and watch lists.

Filters

Lastly, the filter function will also be an important feature of our database. This function will allow users to view animes based on filters. Users will be able to filter from anime genres or anime studios, to to find animes based on their desired specifications. These filters will help group animes for viewing and help users find the right anime. Once filtered, only animes with the chosen genre or studio will appear and the user will be able to traverse through the selection and add animes to lists. This feature will hopefully help users find animes of their preference and grow their anime-watching experience.

Data-Source: MyAnimeList.net

Revised ERD Listed

anime - Anime table to store information regarding an anime series:
PK name varchar(255) NOT NULL,
reviewScore int,
ageRating int,
airedDate DATETIME NOT NULL,
episodes int NOT NULL,
duration int,
summary varchar(1000),
type ('tv', 'movies'),   Disjoint and total
sourceType ('manga', 'lightNovel'),   Overlapping and total
languageType ('sub', 'dub'),   Disjoint and total
imageURL
FK1 directed REFERENCES studios (name)   total one to total one

studios - Studios table holds information about the studio that created an anime:
PK name varchar(255),
specializationType   ('tv', 'movies')   Overlapping and total

streamingPlatform - Streaming platform table allows user to click the URL for the anime:
PK name varchar(255),
platformURL varchar(2083) NOT NULL

animeHasPlatform - Junction Table for handling N:N relationship for anime to platform:
PK id int AUTOINCREMENT,
FK1 animeName REFERENCES anime (name),   total one to total one
FK2 platformName REFERENCES streamingPlatform (name), total one to total one

genre - Table to hold the genre name:
PK genreName varchar(255)

animeHasGenre - Junction table to handle N:N relationship for anime to genre:
PK id int AUTOINCREMENT,
FK1 animeName REFERENCES anime (name)   total one-to-total one
FK2 genreName REFERENCES genre (name)   total one-to-total one

animeList - Table for holding information about an anime list and the type of list it is:

PK id int AUTOINCREMENT,
name varchar(255) NOT NULL,
type (‘recommendedToUser’, ‘watchList’, ‘previouslyWatched’)   Disjoint and total

animeListItem - Junction table to hold information about specific animes in the anime list:
PK id int AUTOINCREMENT,
listIndex int NOT NULL,
FK1 listID REFERENCES animeList (id),   total one to total one
FK2 animeName REFERENCES anime (name)   total one to total one

user - Table to hold login information for all users and their according lists:
PK username varchar(255) NOT NULL UNIQUE,
email varchar(255) NOT NULL UNIQUE,
password varchar(255) NOT NULL,
FK1 recommendedToUser REFERENCES animeList (id)   total one to total one
FK2 watchListREFERENCES animeList (id)   total one to total one
FK3 previouslyWatched REFERENCES animeList (id)   total one to total one

Tables

anime(name: [PK], reviewScore, ageRating, airedDate, episodes, duration, summary,
type: ['tv', 'movies'], sourceType: ['manga', 'lightNovel'], languageType: ['sub', 'dub'], imageURL)
FK1 {directed} references {studios.name},

studios(name: [PK], specializationType: ['tv', 'movies'])

streamingPlatform(name: [PK], platformURL)

animeHasPlatform(id: [PK])
FK1 {animeName} references {anime.name}
FK2 {platformName} references {platform.name}

genre(genreName: [PK])

animeHasGenre(id: [PK])
FK1 {animeName} references {anime.name}
FK2 {genreName} references {genre.name}

animeList(id: [PK], name, type: ['recommendedToUser', 'watchList', 'previouslyWatched'])

animeListItem(id: [PK], listIndex)
FK1 {listId} references {animeList.id}
FK2 {animeName} references {anime.name}

user(username: [PK], email, password)
FK1 {recommendedToUser} references {animeList.id},
FK2 {watchList} references {animeList.id}
FK3 {previouslyWatched} references {animeList.id}

Functional Dependencies

anime(name)

F = {name → reviewScore, ageRating, airedDate, episodes, duration, summary, type, sourceType, languageType, directed, imageURL}

CK = {{name}}

In 4NF

studios(name)

F = {name → specializationType}

CK = {{name}}

In 4NF

streamingPlatform(name)

F = {name → platformURL}

CK = {{name}}

In 4NF

animeHasPlatform(id)

F = {id → animeName, platformName}

CK = {{id}}

In 4NF

genre(genreName)

F = {genreName → ∅}

CK = {{genreName}}

In 4NF

animeHasGenre(id)

F = {id → animeName, genreName}

CK = {{id}}

In 4NF

animeList(id)

F = {id → name, type}

CK = {{id}}

In 4NF

animeListItem(id)

F = {id → listIndex, listID, animeName}

CK = {{id}}

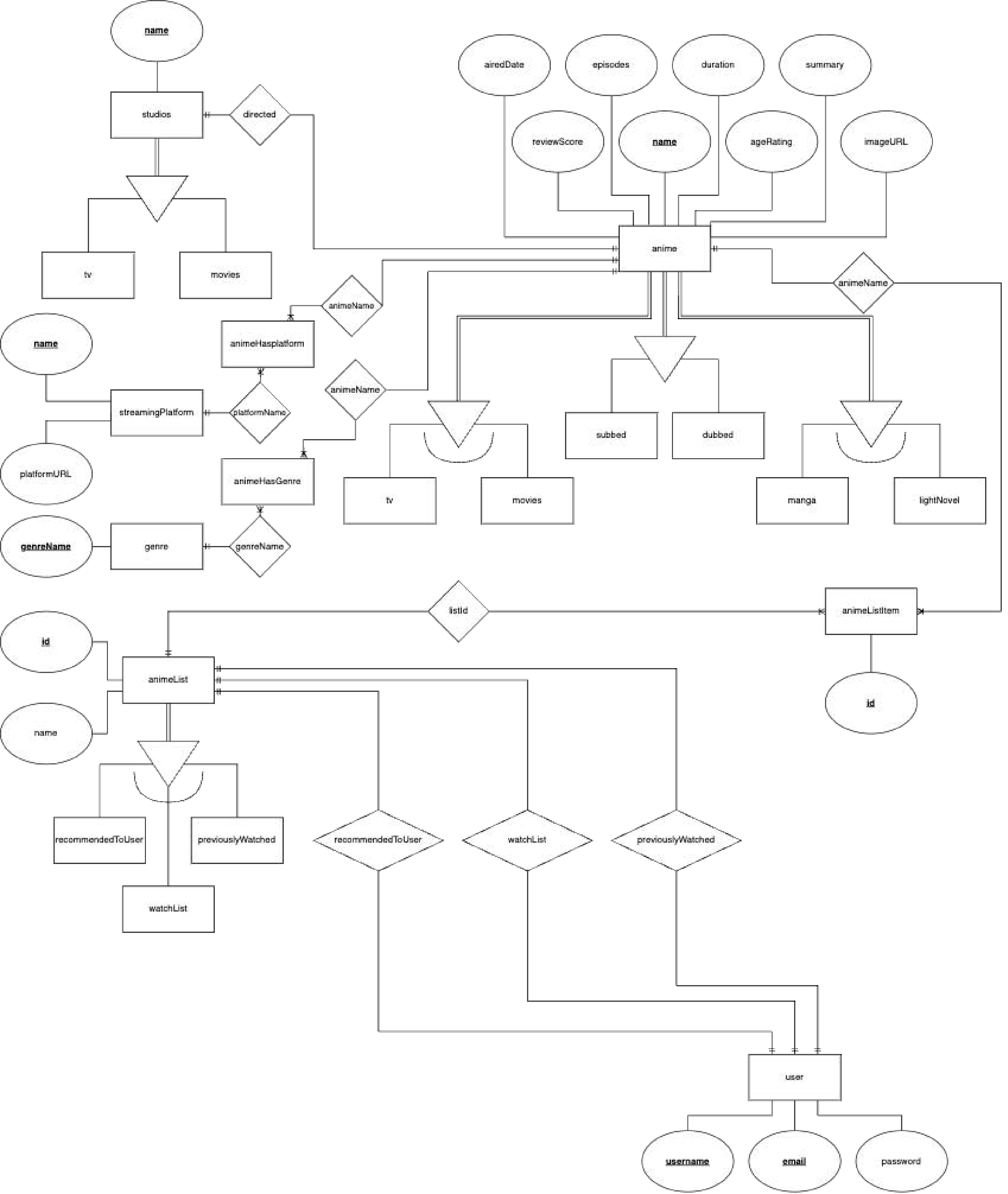
In 4NF

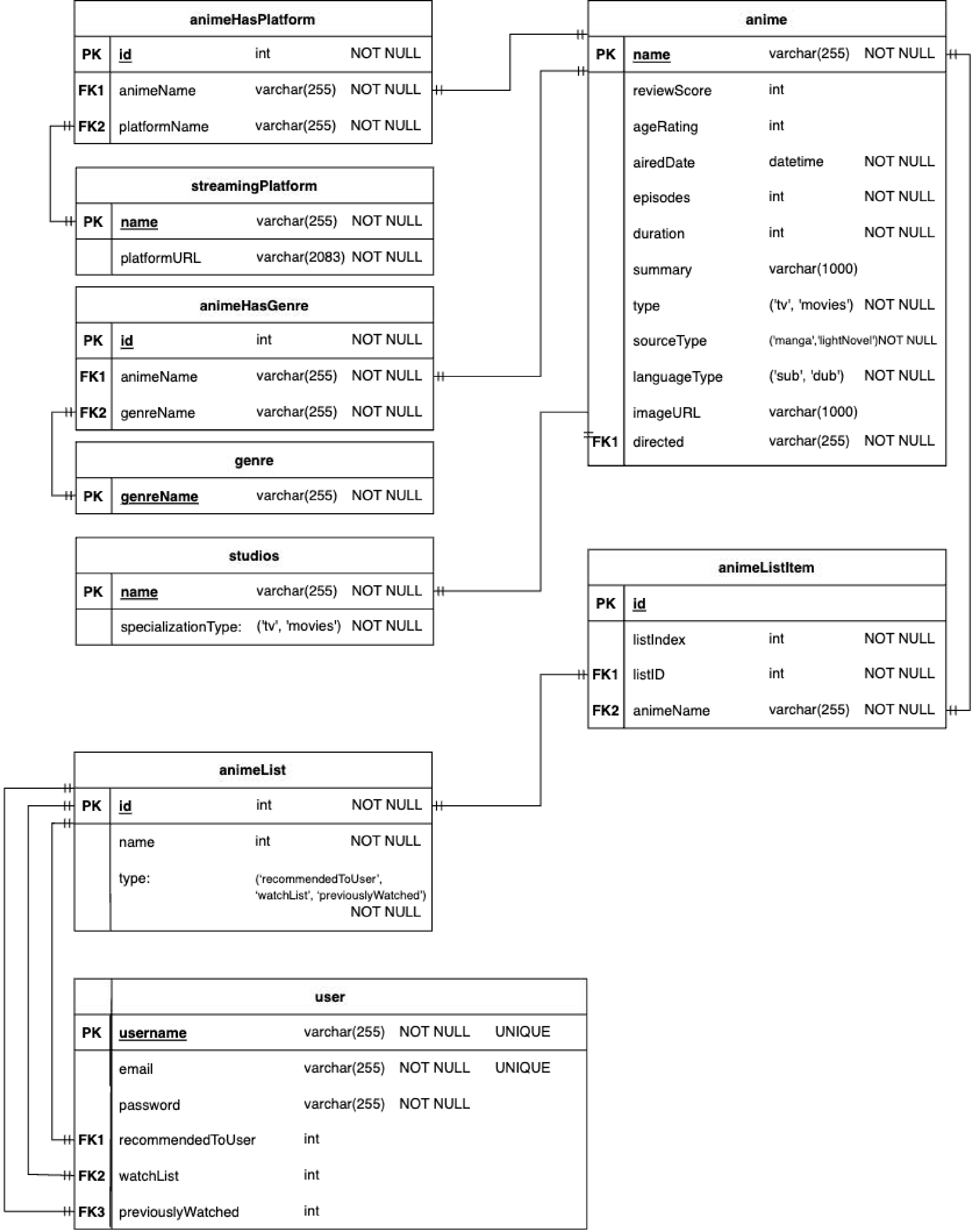
user(username)

F = {username → email, password, recommendedToUser, watchList, previouslyWatched}

CK = {{username}}

In 4NF


name
studios
directed
tv
movies
airedDate
episodes
duration
summary
reviewScore
name
ageRating
imageURL
anime
animeName
animeName
animeHasplatform
name
streamingPlatform
platformName
platformURL
animeName
animeHasGenre
genreName
genre
genreName
tv
movies
subbed
dubbed
manga
lightNovel
listId
animeListItem
id
id
animeList
name
recommendedToUser
previouslyWatched
watchList
recommendedToUser
watchList
previouslyWatched
user
username
email
password



	animeHasPlatform		
PK	id	int	NOT NULL
FK1	animeName	varchar(255)	NOT NULL
FK2	platformName	varchar(255)	NOT NULL

	streamingPlatform		
PK	name	varchar(255)	NOT NULL
	platformURL	varchar(2083)	NOT NULL

	animeHasGenre		
PK	id	int	NOT NULL
FK1	animeName	varchar(255)	NOT NULL
FK2	genreName	varchar(255)	NOT NULL

	genre		
PK	genreName	varchar(255)	NOT NULL

	studios		
PK	name	varchar(255)	NOT NULL
	specializationType:	('tv', 'movies')	NOT NULL

	anime		
PK	name	varchar(255)	NOT NULL
	reviewScore	int	
	ageRating	int	
	airedDate	datetime	NOT NULL
	episodes	int	NOT NULL
	duration	int	NOT NULL
	summary	varchar(1000)	
	type	('tv', 'movies')	NOT NULL
	sourceType	('manga','lightNovel')NOT NULL	
	languageType	('sub', 'dub')	NOT NULL
	imageURL	varchar(1000)	
FK1	directed	varchar(255)	NOT NULL

	animeListItem		
PK	id		
	listIndex	int	NOT NULL
FK1	listID	int	NOT NULL
FK2	animeName	varchar(255)	NOT NULL

	animeList		
PK	id	int	NOT NULL
	name	int	NOT NULL
	type:	('recommendedToUser', 'watchList', 'previouslyWatched')	NOT NULL

	user			
PK	username	varchar(255)	NOT NULL	UNIQUE
	email	varchar(255)	NOT NULL	UNIQUE
	password	varchar(255)	NOT NULL	
FK1	recommendedToUser	int		
FK2	watchList	int		
FK3	previouslyWatched	int		