vYoutube Entity-Relationship Model

Tomás Alejandro Delgado Ortiz

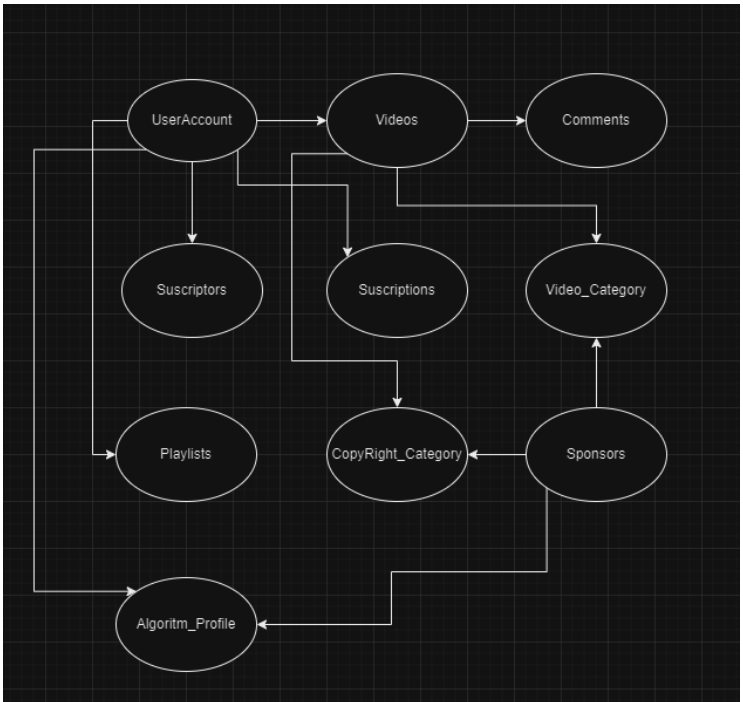
Code: 20221020045

User Stories

- As an User, i want to have an interface where i can find my liked and saved videos, suscriptions and playlists in an organized structure, so i can easily move into the content i want to watch
- As a Content Creator User, i want to have the capability to manage my shared videos satitistic data, in retribution of the comments from my suscriptors community, so i can create more accurate content to my community's requests
- As a User, i want to recieve personalized video recomendations based on my watching history, so that i can discover new content of my interest in the platform
- As an User, i want to be able to create and manage different accounts in the platform, so i can watch different type of content recomendations depending in wich session i´m logged in
- As a Sponsor, i want to be able to check the category, watching statistics and CopyRight category linked to each video, so i can be more accurate promoting my products acording to user needs.
- As a Content Creator User, i want to be able to manage my video´s category and CopyRight Category, so i can create videos in wich the sponsors are interested to monetize

Entity-Relationship Model Design & Creation:

1) Component Definition:
The components are defined and give context about what you want to model.



2) Define Entities: Model the actors that establish a relationship with each other according to the context and what was previously defined.

3) Define attributes by entity: being clear about the fields in which intrinsic data is stored according to the context

- E1 UserAccount: nickname, id email, suscriptions, videos, playlists, suscriptors, Algoritm_Profile
- E2 Video:id, name, date, likes, dislikes, category, Coýright_Category, Author, views, Sponsor, comments, playlist
- E3 Comment: autor, id, content, likes, dislikes, video
- E4 Suscriptor:id, UserAccount, date
- E5 Suscription: id, User Account, date
- E6 Video_Category: name, videos, likes, dislikes
- E7 Playlists: name, User Account, videos, id
- E8 Copyright_Category: name, videos
- E9 Sponsors: name, video, product, Video_Category
- E10: Algoritm_Profile: UserAccount, id, videos

4) Define Relationships between entities according to their functionality within the project

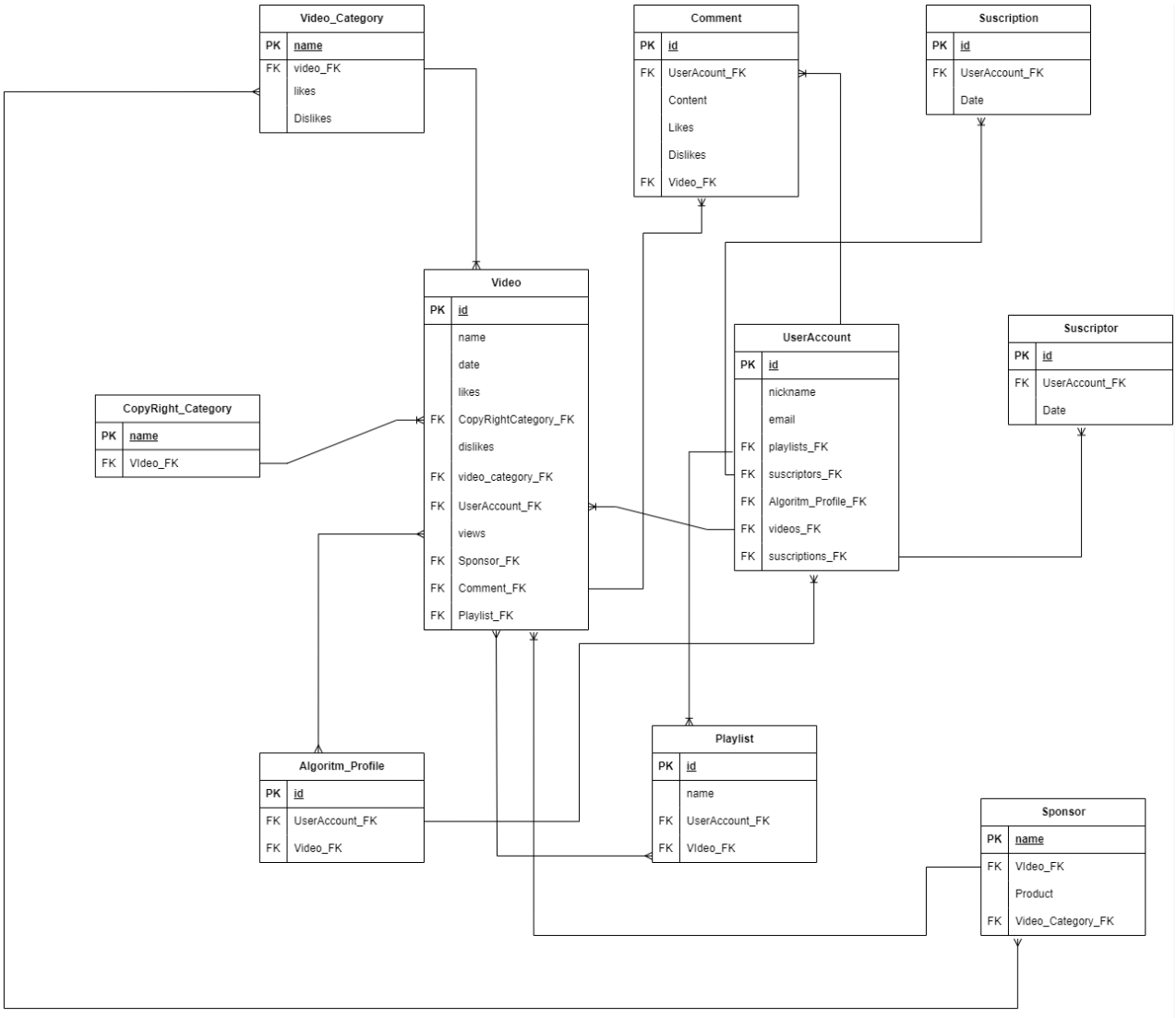
Rel.	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
E1	[redacted]	x		x	x		x			x
E2	x	[redacted]	x			x	x	x	x	x
E3	x	x	[redacted]							x
E4	x			[redacted]						
E5	x				[redacted]					
E6		x				[redacted]	x		x	
E7	x	x					[redacted]			
E8		x						[redacted]	x	
E9		x				x			[redacted]	
E10	x	x								[redacted]

5) Define the types of relationships between entities.

- E1 -one to many- E2
- E1 -one to many – E4
- E1 – one to many- E5
- E1 – one to many – E7
- E1 – many to many – E10
- E2 – one to many – E3

- E6 – one to many – E2
- E2 - many to many – E7
- E8 – one to many – E2
- E2 – many to many – E10
- E1 -one to many – E3
- E3 – one to many- E10
- E6 – many to many – E7
- E9 – many to many – E6
- E9 – one to many -E2

6) First Entity -Relationship Draw: the relationship is drawn similar to a class diagram, highlighting only the relationships between each entity, evidenced in the previous step.



7). Identify primary and foreign keys and use them to break Many - to - many relationships.

Consideration: To break many-to-many relationships, an intermediate entity is created that will have a many-to-one relationship with each end of the relationship, this entity contains the foreign keys of both entities.

8) Second Entity-Relationship Draw: a second drawing is made of the relationships between updated entities

9) Create the E-R-M Data Structure, the data with its respective type is indicated for each of the entities and atributes are asigned to each of entity values