Channels

PK	channel_id
	channel_name

Videos

PK	video_id
	video_title
	view_count
	like_count
	comment_count
	published_date
	trending_date
	link
	thumbnail_link

Publishes

Users

PK	user_id
	user_name
	user_email
	user_password

Favorites

favorite_date

Likes

favorite_date

Category

Categories

PK	category_id
	category_description

Associates

Tags

PK	tag_id
	tag_description

Relations

Users(user_id: *INT* **[PK]**,
      user_name: *VARCHAR(255)*,
      user_email: *VARCHAR(320)*,
      user_password: *VARCHAR(100)*)

Videos(video_id: *VARCHAR(255)* **[PK]**,
      video_title: *VARCHAR(255)*,
      view_count: *INT*,
      comment_count: *INT*,
      like_count: *INT*,
      category_id: *INT* **[FK to Categories.category_id]**,
      channel_id: *INT* **[FK to Channels.channel_id]**,
      uploaded_date: *DATE*,
      trending_date: *DATE,*
      link: *VARCHAR(512)*,
      thumbnail_link: *VARCHAR(512)*)

Categories(category_id: *INT* **[PK]**,
      category_description: *VARCHAR(255)*)

Channels(channel_id: *INT* **[PK]**,
      channel_name: *VARCHAR(255)*)

Tags(tag_id: *INT* **[PK],**
      tag_description: *VARCHAR(255)*)

Favorite_Videos(user_id: *INT* **[PK, FK to Users.user_id]**,
      video_id: *VARCHAR(255)* **[PK, FK to Videos.video_id]**,
      favorite_date: *DATE*)

Favorite_Tags(user_id: *INT* **[PK, FK to Users.user_id]**,
      tag_id: *INT* **[PK, FK to Tags.tag_id]**,
      favourite_date: *DATE*)

Video_Tags(video_id: *INT* **[PK, FK to Videos.video_id]**,
      tag_id: *INT* **[PK, FK to Tags.tag_id]**)

Description & Assumptions

We have 5 entity sets in our ER diagram, namely Videos, Tags, Channels and Categories. The first 4 entities can be found in the Kaggle dataset, while Users is maintained by our web application.

An Users entity is inserted when a new account is created. It stores the user's login information, e.g. his/her name, email address and password. Every user registered in our website corresponds to one of the Users records.

Videos is our core entity set, as it relates to all 4 other entity sets. Since we want to create a more flexible search system, we aim to include as much information as possible from the original Kaggle dataset, such as view_count, comment_count, like_count, upload_date, etc. It also satisfies the referential integrity to Channels or Tags, having fields channel_id and tag_id which are foreign keys to Channels.channel_id and Tags.tag_id respectively.

Tags is made a separate entity set because it is a key feature. Every tag is identified by its tag_id and has tag_name. When a new account is created, the user is prompted to choose which tags which he/she is interested in. Every video is tagged to one or more tags, which enables our search system to recommend videos to the user's liking. Hence Tags has relationships with both Users and Videos.

In the original dataset, every video is from a category and uploaded by a channel. Hence we created entity sets Categories and Channel to store information about these two fields. Every category is identified by category_id and has category_description, while every channel is identified by channel_id and has channel_name.

Many-to-Many relationship between Users and Videos/Tags

The biggest functionality of our project is users being able to curate a playlist of their favorite videos and tags. Hence every user can favorite 0 or more videos and tags. Every video or tag can be favorited by 0 or more users.

Many-to-Many relationship between Videos and Tags

Every tag is tagged to 1 or more videos, but every video has at least 1 tag ([None] is a valid tag in the Kaggle dataset).

One-to-Many relationship between Videos and Categories

Every video is published under 1 category, but every category can have 1 or more videos.

One-to-Many relationship between Channels and Videos

Every channel has published 1 or more YouTube videos, but every video is published by 1 channel.