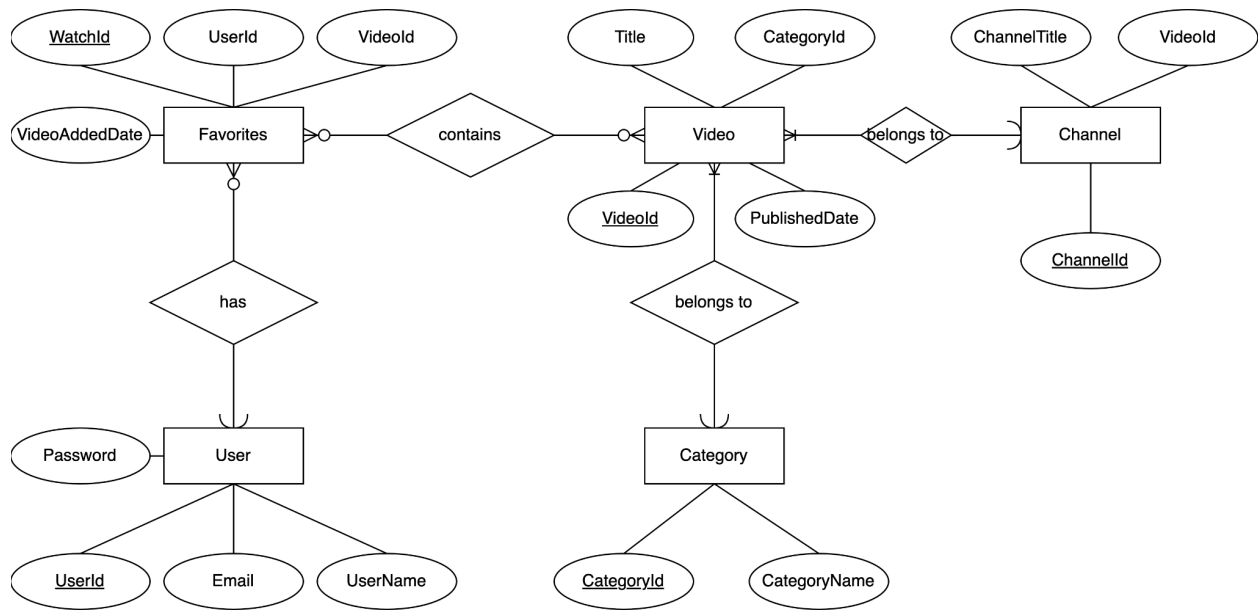# ER Diagram



# Relational Schema Definition

User(
    UserId:INT [PK],
    UserName:VARCHAR(100),
    Email:VARCHAR(255),
    Password:VARCHAR(255)
)

Favorites(
    WatchId:INT(PK),
    UserId:INT [FK to User.UserId],
    VideoId:CHAR(11),
    VideoAddedDate: DATE
)

Video(
    VideoId:CHAR(11)[PK],
    TItle:VARCHAR(100),
    CategoryId:CHAR(2),
    PublishedDate: DATE
)

Category(
      CategoryId:CHAR(2)[PK] ,
      CategoryName:VARCHAR(100)
)

Channel(
      ChannelId:CHAR(24) [PK],
      ChannelTitle:VARCHAR(255),
      VideoId:CHAR(11)[FK to Video.VideoId]
)

# Description and Assumptions of Relations:

In general, we aim to develop a hot video recommendation application based on the dataset of daily trending YouTube Videos. The above relational scheme consists of five tables: User, Favorites, Video, and Category, as well as Channel. We aligned our schema data types using the dataset listed types and the standard Youtube character regulation limitations.

- **User**
    - **Description:** The User table stores user information, including their unique UserId, UserName, Email, and Password. The UserId serves as the primary key of the User table.
    - **Assumption:** We proposed that every user could have 0 to many favorite lists and one favorite list can only be associated with one user.
- **Favorite**
    - **Description:** The Favorites table stores user favorites, with each record consisting of a unique WatchId, the corresponding UserId from the User table (via a foreign key), the VideoId of the favorite video, and the date the video was added. The WatchId serves as the primary key of the Favorites table.
    - **Assumption:** Each favorite list may contain 0 or many videos, and a single video can be liked by many users,  thus one video can belong to 0 or many favorite lists.
- **Video**
    - **Description:** The Video table stores information about each video, including its unique VideoId, the Title of the video, the CategoryId of the video (which connects to the Category table), and the date the video was published. The VideoId serves as the primary key of the Video table.
    - **Assumption:** Further analysis of the dataset revealed that each video has a single category id, but a category can contain multiple videos.
- **Category**
    - **Description:** The Category table stores information about each video category, including its unique CategoryId and CategoryName. The CategoryId serves as the primary key of the Category table.
    - Assumption: As stated before, each category could contain multiple videos, but one video can only belong to one category.

- **Channel**
  - **Description:** The Channel table stores information about each video channel, including its unique ChannelId, the ChannelTitle, and the VideoId of the video (via a foreign key to the Video table). The ChannelId serves as the primary key of the Channel table.
  - **Assumption:** Each video belongs to only one channel, but a channel can contain 1 to many videos.