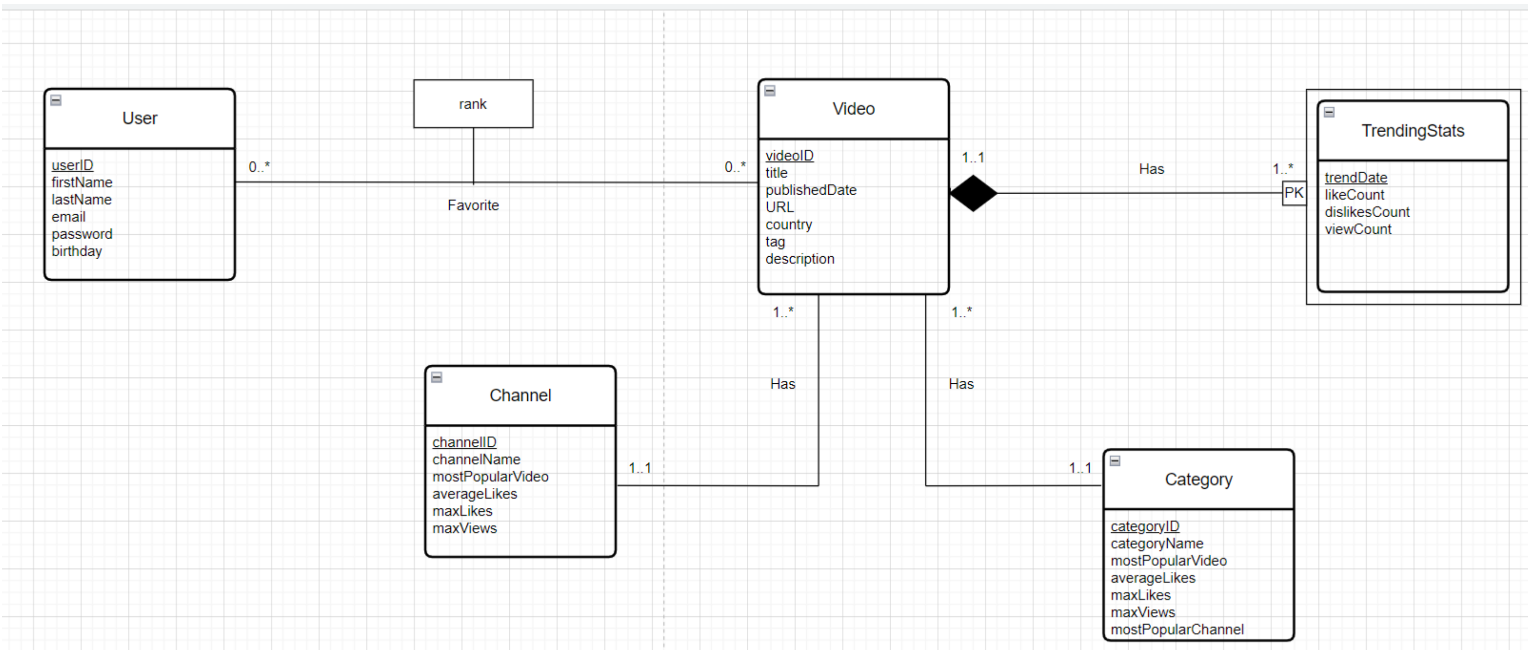
TubeTrendz Database UML Diagram:



Database Normalization:

Let relation R define our Database:

Functional Dependencies = {

userID -> firstName, lastName, email, password, birthday
userID, videoID -> rank
userID -> videoID
videoID -> title, publishedDate, URL, country, tag, description
videoID -> trendDate
trendDate -> likeCount, dislikesCount, viewCount
videoID -> categoryID
categoryID -> categoryName, mostPopularVideo1, averageLikes1, maxLikes1, maxViews1, mostPopularChannel
videoID -> channelID
ChannelID -> channelName, mostPopularVideo2, averageLikes2 maxLikes2, maxViews2
mostPopularVideo1 -> videoID
mostPopularVideo2 ->videoID
mostPopualurChannel -> ChannelID

}

Decomposing R into 3NF:

1. Find Minimal basis of R:
    a. Remove unnecessary attributes from LHS
        i. userID,videoID -> Rank

UserID videoID → Rank

Try userID → Rank

$userID^+$ = { every attribute except Rank }

$videoID^+$ = { every attribute except Rank, UserID, FirstName, lastName, email, password birthday }

   ii. Since userID and videoID cannot reach attribute Rank by closure of userID or closure of videoID functional, we cannot simplify the LHS of this FD any further.

b. Remove FDs that can be inferred from the rest

   i. None of the FDs listed above can be removed from the relation as they cannot be inferred from the rest.

2. For each FD in the minimal basis, use the attributes in the FD to define the schema for the new relation.
   a. Users (userID, firstName, lastName, email, password, birthday)
   b. Favorites(userID, videoID, rank)
   c. Videos(videoID, title, publishedDate, URL, country, tag, description)
   d. Channel(channelID, channelName, mostPopularVideo2, averageLikes2, maxLikes2, maxViews2)
   e. Category(categoryID, categoryName, mostPopularVideo1, averageLikes1, maxLikes1, maxViews1, mostPopularChannel)
   f. Pop(mostPopularVideo1, videoID)
   g. Pop2(PopularVideo2, videoID)
   h. Pop3(mostPopularChannel, ChannelID)
3. The schema labeled Favorites(userID, videoID, rank) includes a superkey as you can reach all attributes in the database from attributes userID and videoID. Since a superkey is included in the database already, the definition of our database satisfies the conditions for 3NF.

Our schema as illustrated above demonstrates our database follows 3NF. We chose 3NF over BCNF, because 3NF conserves functional dependency while BCNF does not during decomposition. 3NF is also less strict compared to BCNF which makes it easier to implement.

Database Relational Schema:

<u>Relational Schema</u>

User(userID: INT [Primary Key], FirstName: VARCHAR(MAX), LastName: VARCHAR(MAX), Email: VARCHAR(MAX), Password: VARCHAR(MAX), Birthday: DATE)

Video(videoID: VARCHAR(MAX) [PK], channelID: VARCHAR(MAX) [FK to Channel.channelID], categoryID: INT [FK to Category.categoryID], title: VARCHAR(100), publishedDate: DATE, URL: VARCHAR(MAX), country: VARCHAR(MAX), tag: VARCHAR(MAX), description: VARCHAR(5000), trendDate: DATE [FK to TrendingStats.trendDate)

Favorite(userID: INT [FK to User.userID], videoID: VARCHAR(MAX) [FK to Video.videoID], rank: INT)

Channel(channelID: VARCHAR(MAX) [Primary Key], channelName: VARCHAR(MAX), mostPopularVideo2: VARCHAR(MAX) [FK to Video.videoID], averageLikes2: INT, maxLikes2: INT, maxViews1: INT)

Category(categoryID: INT [PK], categoryName: VARCHAR(MAX), mostPopularVideo1: VARCHAR(MAX) [FK to Video.videoID], averageLikes1: INT, maxLikes1: INT, maxViews1: INT, mostPopularChannel: VARCHAR(MAX) [FK to Channel.channelID])

Relationship Descriptions

1. User ⇔ Video
    a. Many-to-Many relationship
    b. A user can have multiple favorite videos, and each video can be a favorite of multiple users
2. Video ⇔ Category
    a. 1-to-Many relationship
    b. Each video has one category, each category can have multiple videos
3. Video ⇔ Channel
    a. 1-to-Many relationship
    b. Each video has one channel, each channel can have multiple videos
4. Video ⇔ TrendingStats
    a. Weak entity (TrendingStats)
    b. 1-to-Many relationship
    c. Each video can have multiple trending stats (stats are by date)