A diagram of a diagram

Description automatically generated with medium confidence

Description and Assumption

Entity

User: Generated for anyone who wishes to use TuberInsights. Users must first provide userName and password to register. UserId will be assigned to each user during registration.

Normal User: A subclass from User. This will be the default type for most users as these will be the customers trying to require predictions and view trending video info. UserType will be marked as a normal user when registering.

Admin: A subclass from User. This will be the user type for website administrators who can update and delete video and channel information.

Prediction: A weak entity referencing users. Normal users can request predictions about suggested tags for their videos by inputting some descriptions about their videos. When requesting a prediction, a new row will be added to the prediction table with the user ID, input, output, and unique prediction ID.

channel: Each Youtuber has its own channel. In the video table, each row will have a channel ID referencing the channel table. The channel table provides channel title paring with channel ID.

video: All info provided by the raw data is stored in this table. Besides, based on different CSV files of different locations, each row will have a region attribute. video reference to channel table by channel ID and reference to category table by category ID.

category: a table stores info about the category ID and its corresponding category name.

Relationship

User-Prediction Relationship (Request):

* One-to-Many from User to Prediction
* Assumptions: A user can request multiple predictions, but each prediction is associated with only one user.

User-Video Relationship (Update):

* Many-to-Many between User and Video through the Update entity
* Assumptions: Users can update multiple videos, and videos can be updated by multiple users.

Channel-Video Relationship (Publish):

* One-to-Many from Channel to Video
* Assumptions: A channel can publish multiple videos, but each video belongs to one and only one channel.

Category-Video Relationship (Belong):

* One-to-Many from Category to Video
* Assumptions: A category can have multiple videos, but each video belongs to one and only one category.

Normalization

To begin with, our team choose to use 3NF to normalize our scheme. Comparing with BCNF, 3NF will keep some related dependencies in one table which will lead to simpler table implementations. For our database design, since our database do not have very complicated dependencies (basically for each entity, all attributes depend on the prime key of the entity), we decided to use 3NF to make our design tide and workable.

User(userId:INT [PK], userName:VARCHAR(30), password:VARCHAR(40), userType ENUM('admin', 'default'))  
  
  
Prediction(predictId:INT [PK], userId: INT [PK] [FK to User.userId], model ENUM('tag\_generation', 'trending\_prediction'), input:VARCHAR(100), result:VARCHAR(100))  
  
  
Channel(channelId: INT [PK], channelTitle VARCHAR(30))  
  
  
Update(video\_id: INT [PK] [FK to Video.video\_Id],  userId: INT [PK] [FK to User.userId)  
  
  
Category(categoryId: INT [PK], categoryName: VARCHAR(30))  
  
  
Video(video\_id INT [PK], channelId: INT  [FK to Channel.channelId], categoryId: INT [FK to Category.categoryId], dislikes: INT, likes: INT, description: TEXT, tags: VARCHAR(255), trending\_date: DATE, title: VARCHAR(255), comment\_count: INT, view\_count: BIGINT, published\_at: DATE, Region: VARCHAR(255))

This is our current relational schema translated from the ER diagram. It has already met the requirements of 3NF.

For the user table, dependency indicates that username, password and user type are all dependent on user id.

For prediction, since prediction is a weak entity, so prediction id and user id together decide output and model. Even though prediction id depends on user id and input, prediction id is part of the prime key for this table, so 3NF still stands.

For channel table and category table, channel title depends on channel id and category name depends on category id so these tables meet 3NF requirements.

For update table, this table only store any maintenance record of admin updating video table.