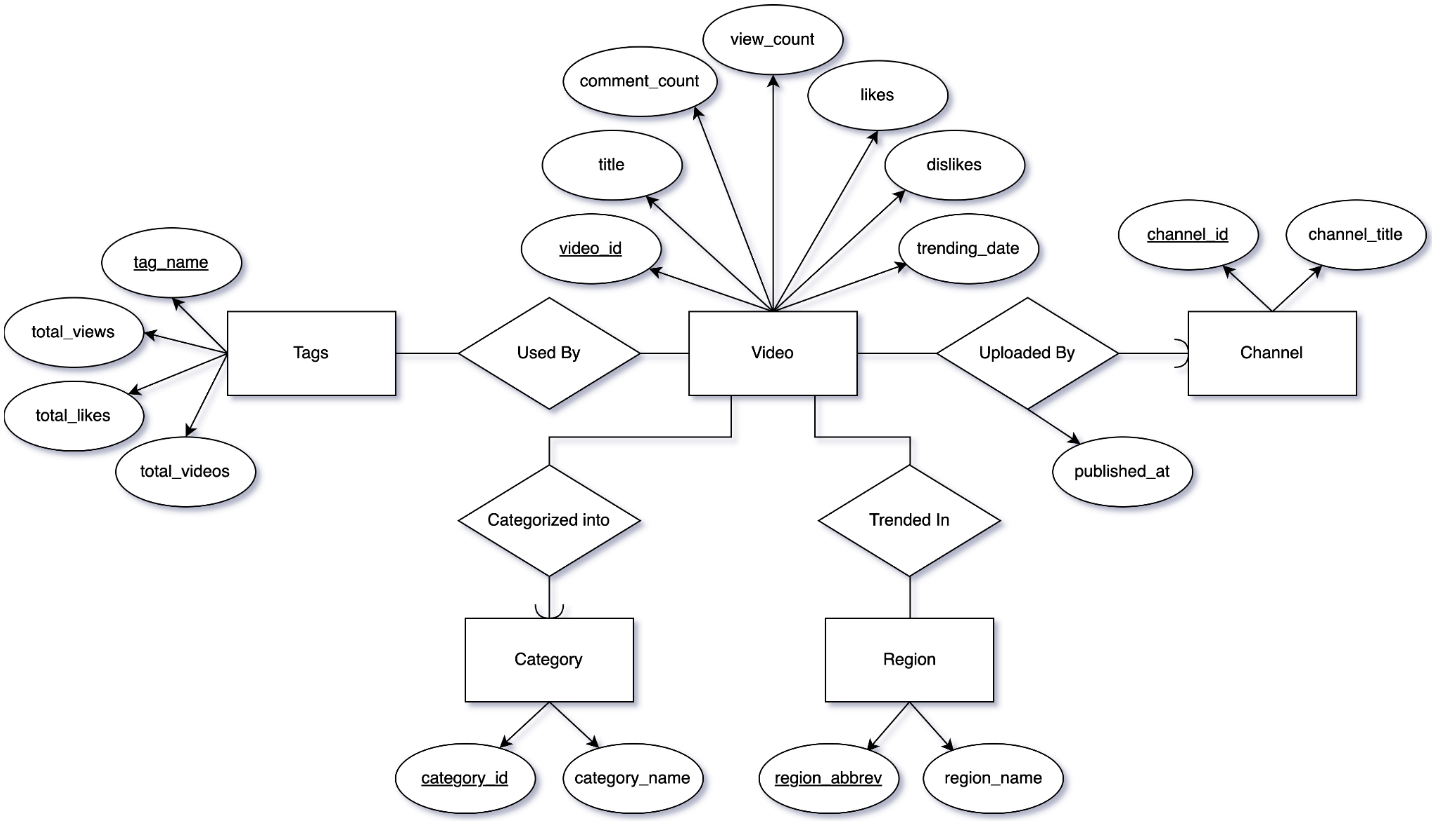
Database Design



Our Assumptions for each entity-entity relationship:

Video-Channel (many-1): We think each video must only belong to one channel, but one channel can upload many trending videos.

Video-Region (many-many): Since the same video can appear in multiple region-distinguished datasets, we assume each video can trend in one or more regions. Each region will have many videos that trend within it.

Video-Category (many-1): We assume that each video can be categorized into only one category of 32. However, each category is assumed to have many videos under it.

Video-Tag (many-many): When creators upload their videos, they have the option to add any number of desired tags, and the same tag can appear under any number of videos. Thus the assumptions we made are that each video can have many tags and that each tag can belong to one or more videos.

Functional Dependencies and Following BCNF:

tag_name-> total_views, total_likes, total_videos
category_id->category_name
region_abbrev->region_name
channel_id->channel_title
video_id->title, trending_date, dislikes, likes, view_count, comment_count, published_at

These relations follow BCNF, because for each of the relations the superkey is on the left, and it covers all the attributes in the relation. When we group each of the relations, we see that all attributes in the entities get covered because the primary key has a relation to each of the attributes in the table. The given relation also follows 3NF, since the left part is a key.

Logical Design (Relational Schema):

Channel(channel_id: VARCHAR(255) [PK], channel_title: VARCHAR(255))

Video(video_id: VARCHAR(255) [PK], title: VARCHAR(255), trending_date: DATETIME, channel_id: VARCHAR(255) [FK to Channel], comment_count : INT, view_count : INT, likes : INT, dislikes : INT, published_at: DATETIME, category_id: INT[FK to Category], tags: VARCHAR(255))

Tags(tag_name: VARCHAR(255)[PK], total_views : INT, total_likes : INT, total_videos : INT)

Category(category_id: INT[PK], category_name: VARCHAR(255))

Region(region_abbrev: VARCHAR(255) [PK], region_name: VARCHAR(255))

Used_By(tag_name: VARCHAR(255) [PK], video_id: VARCHAR(255) [PK])

Trended_In(region_abbrev: VARCHAR(255)[PK], video_id: VARCHAR(255) [PK])