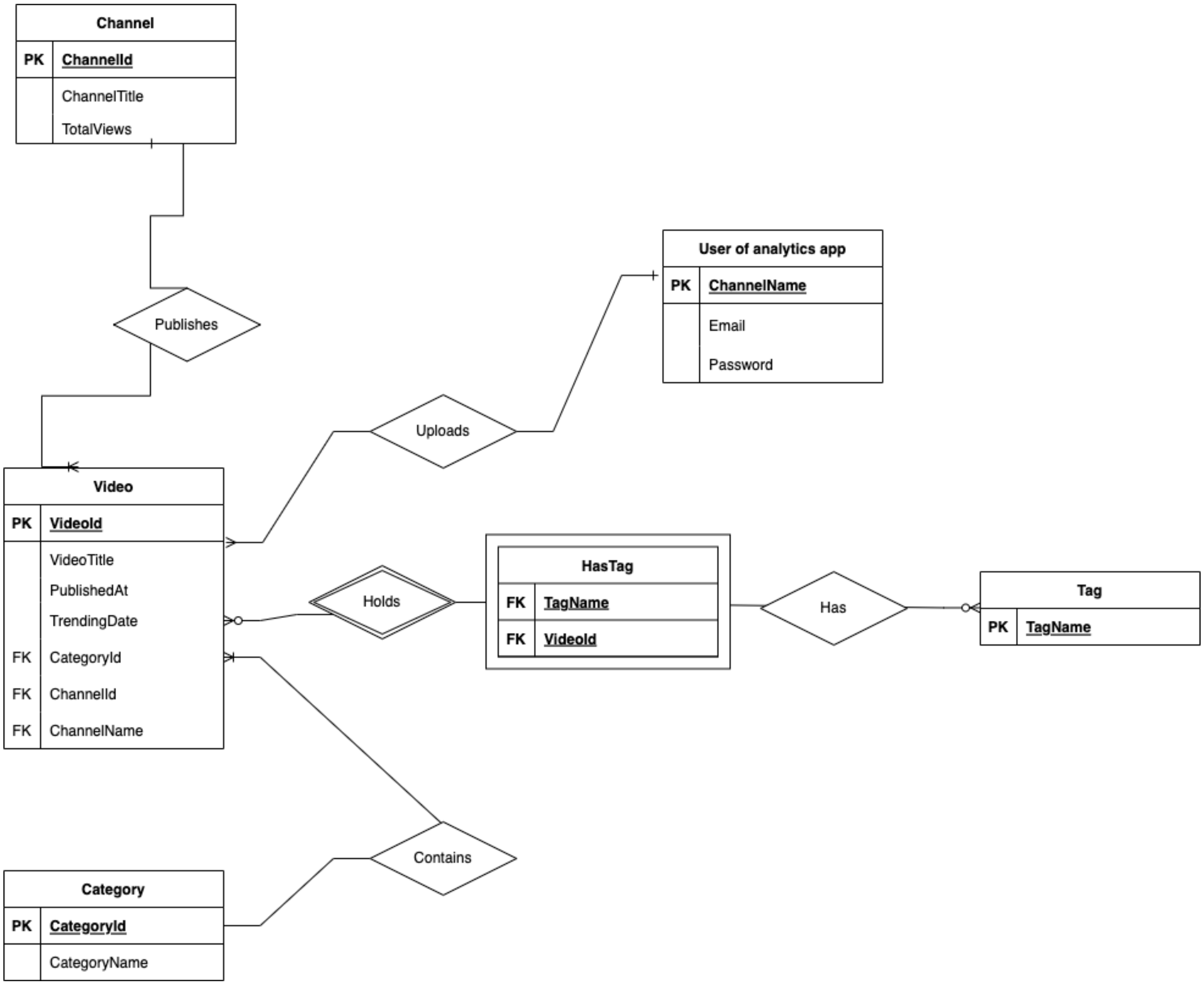
Project Stage 2 ER diagram



Description of Assumptions:

The video entity contains a primary key called video_id and also contains other attributes such as title, publishedAt, trendingDate. Video is the entity with the most relations. For the video to category relationship, we assume that every video can have a maximum of one category but every category can have multiple videos. Category entity has categoryId as its primary key.

For video to tags relationship, we assume that every video can have 0 to multiple tags as well as every tag can have 0 to multiple videos. The tag just has the primary key variable Name as its key.

We are creating a weak entity has tag which has two foreign keys video_id and tag_name which acts as a relationship table between video and tags.

The user of analytics app entity has one primary key channel name and other attributes such as email, password for login to the web app. For user of analytics entity, we assume a user can upload many videos.

The channel entity has its primary key as channelId with other attributes such as channel title, total views of the channel. For the channel to video entity, we assume a channel can have many videos so it is a 1 to many relationship.

Relational Schema

Channel(channelId: VARCHAR(255) [PK], channelTitle: VARCHAR(255))

User(channelName: VARCHAR(255) [PK], Email: VARCHAR(255), Password: VARCHAR(255))

Video(Video_id: VARCHAR(255) [PK], title: VARCHAR(255), pulishedAt: DATETIME, trendingDate: DATETIME, categoryId: INT [FK to Category], channelId: VARCHAR(255) [FK to Channel], channelName: VARCHAR(255) [FK to User])

HasTag(Video_id: VARCHAR(255) [FK to Video], Tag_name: VARCHAR(255) [FK to Tag])

Tag(Name: VARCHAR(255)[PK])

Category(categoryId: INT[PK], categoryName: VARCHAR(255))

DDL Commands:

```
CREATE TABLE Channel (
   ChannelId VARCHAR(255) NOT NULL,
   ChannelTitle VARCHAR(255),
   TotalViews INT,
   PRIMARY KEY (ChannelId)
);

CREATE TABLE User (
   ChannelName VARCHAR(255) NOT NULL,
   Email VARCHAR(100),
   Password VARCHAR(255),
   PRIMARY KEY (ChannelName)
);


CREATE TABLE Video (
   VideoId VARCHAR(255) NOT NULL,
   VideoTitle VARCHAR(255),
   CategoryId  INT NOT NULL,
   ChannelId VARCHAR(255) NOT NULL,
   ChannelName VARCHAR(255) NOT NULL,
   PRIMARY KEY (VideoId),
   FOREIGN KEY (CategoryId) REFERENCES Category(CategoryId),
   FOREIGN KEY (ChannelId) REFERENCES Channel(ChannelId),
   FOREIGN KEY (ChannelName) REFERENCES User(ChannelName)
);



CREATE TABLE Tag (
   TagName VARCHAR(255) PRIMARY KEY
);



CREATE TABLE Category (
   CategoryId INT NOT NULL,
   CategoryName VARCHAR(100),
   PRIMARY KEY (CategoryId)
);

CREATE TABLE HasTag (
   VideoId VARCHAR(255) ,
   TagName VARCHAR(255),

   FOREIGN KEY (VideoId) REFERENCES Video ON DELETE CASCADE

   FOREIGN KEY (TagName) REFERENCES Tag ON DELETE CASCADE
);
```