

Milestone 1: Form a Team and Outline a Goal.

1. Write down the names of the people in your group, and your option choice.
2. Describe what data you plan to use, what insights can be derived from this data, and why you think the listed insights are important.
3. Describe how you plan to design the database to store this data and explain your Rationales.
  - Outline the tables that you plan to work with, and describe the column names, their corresponding data types and constraints, and the relationships between the tables.
4. Describe your teamwork: how did you decide on the project goal, list the contribution of each team member.

Options Choice: A

Data Collection:

- Watch History: Collect information about the videos a user has watched, including the video title, URL, date, and time.
- Search Queries: Record the keywords and phrases users use when searching for videos.
- Subscriptions: Keep track of the channels a user is subscribed to and when they subscribe or unsubscribe.
- Likes and Dislikes: Store data on the videos a user has liked or disliked.
- Comments: Save comments made by the user on videos, including the comment text, video title, and timestamp.
- View Duration: Record how long a user typically watches a video before clicking away.
- Video Categories: Categorize videos into genres or topics (e.g., music, gaming, technology) based on video metadata.

Insights:

- Content Preferences: Analyzing a user's watch history, likes, and dislikes can reveal their content preferences. This information is valuable for suggesting personalized video recommendations.

- User Engagement: By looking at the frequency and recency of interactions (watching, liking, commenting), you can gauge how engaged a user is with the platform.
- Popular Content: Determine which types of content are most popular among your users based on their subscriptions and watch history. This can help in tailoring content recommendations.
- User Growth: Tracking subscriptions and unsubscriptions can provide insights into the growth or decline of a user's interest in particular channels or topics.
- Search Behavior: Analyzing search queries can reveal what users are actively looking for on the platform, allowing you to understand trends and improve search functionality.
- Session Length: Understanding how long users typically stay on the platform per session can help optimize content recommendations and user experience.

Information from the Database:

**videos-stats.csv:**

Title (String): Video Title.

Video ID (String): The Video Identifier.

Published At (Date): The date the video was published in YYYY-MM-DD.

Keyword (String): The keyword associated with the video.

Likes (Integer): The number of likes the video received. If this value is -1, the likes are not publicly visible.

Comments (String): The number of comments the video has. If this value is -1, the video creator has disabled comments.

Views (Integer): The number of views the video got.

**comments.csv:**

Video ID (String): The Video Identifier.

Comment (String): The comment text.

Likes (Integer): The number of likes the comment received.

Sentiment (Integer): The sentiment of the comment. A value of 0 represents a negative sentiment, while values of 1 or 2 represent neutral and positive sentiments respectively.

Possible Tables to be created from above:

**Videos Table:**

Columns:

VideoID (Primary Key, Integer): Unique identifier for each video.

Title (String): Title of the video.

URL (String): URL of the video.

CategoryID (Foreign Key, Integer): References the Video Categories table.

Upload\_date (Timestamp): Date and time when the video was uploaded.

Views (Integer): The number of views the video got.

**Likes Table:**

Columns:

LikeID (Primary Key, Integer): Unique identifier for each like.

UserID (Foreign Key, Integer): References the Users table for the user who liked the video.

VideoID (Foreign Key, Integer): References the Videos table for the liked video.

Like\_date (Timestamp): Date and time when the like was made.

**Dislikes Table:**

Columns:

DislikeID (Primary Key, Integer): Unique identifier for each dislike.

UserID (Foreign Key, Integer): References the Users table for the user who disliked the video.

VideoID (Foreign Key, Integer): References the Videos table for the disliked video.

Dislike\_date (Timestamp): Date and time when the dislike was made.

**Comments Table:**

Columns:

CommentID (Primary Key, Integer): Unique identifier for each comment.

CserID (Foreign Key, Integer): potential primary key if a user table is introduced.

Garett Ivie, Justin Bogner, Dominic Frederick, John Turner

VideoID (Foreign Key, Integer): References the Videos table for the commented video.

Comment\_text (Text): Text of the comment.

Comment\_date (Timestamp): Date and time when the comment was posted.

### **Search Queries Table:**

Columns:

QueryID (Primary Key, Integer): Unique identifier for each search query.

UserID (Foreign Key, Integer): References the Users table for the user who performed the search.

Query\_text (String): The search query text.

Query\_date (Timestamp): Date and time when the search query was made.

### **Video Categories Table:**

Columns:

CategoryID (Primary Key, Integer): Unique identifier for each video category.

Category\_name (String): Name of the video category.

**The tables for video-stats and comments have a clear one-to-many relationship because each video can have many comments.**

### **GROUP WORK:**

In conclusion, we decided to go with option A as it's the most relatable category for database collection. We each took part in the milestone and decided to brainstorm possible ideas for tables that could be made or manually generated with relevance to the database above. Garrett did the majority of the writing, Justin did a lot of the brainstorming, Dominic helped out with both writing and brainstorming and John helped with the database information retrieval.