DA Project_Phase_1

Team 20

Introduction

The mini-world is a Youtube-esque ecosystem. It contains data about accounts, videos, comments, ads, etc.

It stores a minimal version of youtube, more reminiscent of early youtube, click here to be taken back to that time.

Purposes

- To monitor the growth of a youtuber/streamer on the platform.
  Such as the number of views and click through rate of a viral video or the statistics of a youtuber.
- To service ads and sponsorships.
  Such as the decision to service 'Raid Shadow Legends' to any and every video game related youtuber.
- To identify bots and spam.
  Such as a user that has over a hundred thousand subscribers but was created 2 minutes ago or to identify fake gift card giveaways in the comments.
- To recommend videos a user likes.
  Such as a user that watches predominantly video game content.

Users

- Ad companies
- Youtubers and streamers
- Third parties looking to use the youtube-api

Applications

Youtube™
To provide the youtube service. Obtain statistics of the site to optimize engagement.

Ad Companies
Can identify which ads are effective and their reach.

Third Parties and Developers
To create custom applications and innovative ideas to extend the capabilities of youtube or integrate youtube within their own applications.

Assumptions

- An account cannot subscribe to itself.
- An ad must run over a video, it cannot be independent.
- A comment only exists for a video, i.e. comments cannot reply to comments.

Entity Types

ENTITY - Strong Entity Type
Entity - Weak Entity Type

Attribute - Key Attribute
[*Attribute*](...) - Composite Attribute
ATTRIBUTE - Multivalued Attribute
Attribute - Partial Key
(Attribute) - Derived Attribute

ACCOUNT.(view_count) is derived by summing over the views of every video the account has uploaded.
ACCOUNT.(revenue) is derived from the individual revenue from each video uploaded by the account.

Entity	Attributes
ACCOUNT	account_id | username | email | description | [*join_time*](seconds, minutes, hours, day, month, year) | subscriber_count | (view_count) | (revenue) | link | watch_time
VIDEO	video_id | title | duration | [*upload_time*](seconds, minutes, hours, day, month, year) | clickthrough_percentage | thumbnail | description | likes | dislikes | rating | content | views | GENRE | revenue
AD	ad_id | advertiser | views | duration | skippable | LOCATION
COMMENT	comment_id | [*comment_time*](seconds, minutes, hours, day, month, year) | likes | dislikes | hearted | pinned | content
Playlist	account_id | title | description | video_count | visibility_status | [*modified_time*](seconds, minutes, hours, day, month, year) | GENRE
Post	account_id | title | description | type | [*post_time*](seconds, minutes, hours, day, month, year)

Relationship Types

PLAY (Degree: 2)

An account plays (or watches) a video.

Participating Entity Types: **ACCOUNT**, **VIDEO**
Cardinality Ratio: (N,M)
(min,max) Constraint: (0,N) **ACCOUNT** PLAYS (0,M) **VIDEO**s

UPLOAD (Degree: 2)

An account uploads a video.

Participating Entity Types: **ACCOUNT**, **VIDEO**
Cardinality Ratio: (1,N)

(min,max) Constraint: **ACCOUNT** (0,N) UPLOADS (1,1) **VIDEO**
Total Participation: Every video must be uploaded by an account.

SUBSCRIBE (Degree: 2)

An account(subscriber) subscribes to another account(subscribed to).

Participating Entity Types: **ACCOUNT**, **ACCOUNT**
Cardinality Ratio: (N,M)
(min,max) Constraint: **ACCOUNT** (0,N) SUBSCRIBEs to (1,M) **ACCOUNT**s
Total Participation: Every subscribed account must have a subscriber

COMMENT (Degree: 3)

An account comments a ‘comment’ on a video.

Participating Entity Types: **ACCOUNT**, **VIDEO**, **COMMENT**
Cardinality Ratio: (L,M,N)
// Read this for (min,max) in a ternary relation
(min,max) Constraint:
ACCOUNT (0,L) COMMENT
COMMENTs (1+1,M) COMMENT
VIDEOs (1,N) COMMENT
Total Participation: Every comment on a video must be by an account.

RUN (Degree: 2)

An advertisement runs over a video.

Participating Entity Types: **AD**, **VIDEO**
Cardinality Ratio: (N,M)
(min,max) Constraint: **AD** (1,N) RUNs on (0,M) **VIDEO**s

CREATE (Degree: 2)

An account creates a playlist.

Participating Entity Types: **ACCOUNT**, *Playlist*
Cardinality Ratio: (1,N)
(min,max) Constraint: **ACCOUNT** (0,N) CREATEs (1,1) *Playlists*
Total Participation: Every playlist is created by an account.

POST (Degree: 2)

An account posts a post.

Participating Entity Types: **ACCOUNT**, *Post*
Cardinality Ratio: (1,N)
(min,max) Constraint: **ACCOUNT** (0,N) POSTs (1,1) *Posts*
Total Participation: Every post is posted by an account.

ENGAGE_WITH (Degree: 4)

An account ENGAGEs WITH a video that had an ad by creating a comment on it.

Participating Entity Types: **ACCOUNT, VIDEO, AD, COMMENT**
Cardinality Ratio: (L,M,N,P)
// Read this for (min,max) in a ternary relation (extended for quaternary)
(min,max) Constraint:
ACCOUNT (0,N) ENGAGE_WITH
VIDEO (0,N) ENGAGE_WITH
AD (1+1,N) ENGAGE_WITH
COMMENT (1+1,N) ENGAGE_WITH
Relationship Attributes:
engagement_rate_ad
percentage of people that commented after watching a video with an advertisement

engagement_rate_no_ad
percentage of people that commented after watching a video without an advertisement

Functional Requirements

1. Retrieval
    A. Query Function
        I. Selection
            - Get all videos uploaded by an account
            - Get all comments commented by an account
            - Get all ads that ran on a particular video
            - Get all posts posted by an account
        II. Projection
            - Get all videos with more than a million views
            - Get all accounts with more than a hundred thousand subscribers
            - Get all comments that have a like-dislike ratio greater than 0.5
        III. Aggregate
            - Total watchtime of an account (SUM)
            - Video with the most and least views on an account (MAX, MIN)
            - Average ad revenue generated by an account in a month (AVG)
        IV. Search
            - Search for an account whose name starts with ‘Pyro’.
            - Search for videos that contain ‘minecraft’ in the title.
    B. Analysis
        I. Generate revenue report earned for each video of an account and also display total revenue earned.
        II. Find out statistics of engagement for videos that have an advertisement, and what kind of advertisements.
        III. Find the like dislike ratio of a certain account’s videos.
2. Modification
        I. Upload a new video, and check if the title is null (INSERT)
        II. Update the title of a video (UPDATE)
        III. Delete a comment on a video (DELETE)