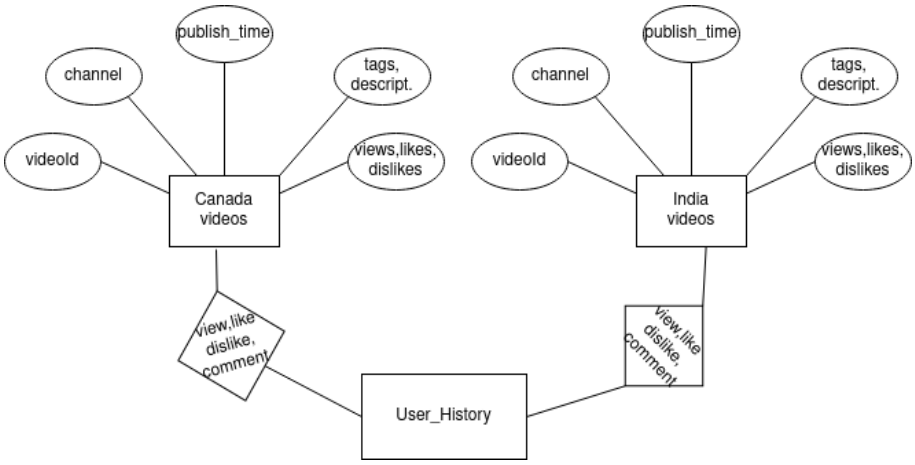
COL362::Project::XploreTube

Sidharth Agarwal 2019CS50661

February 27, 2022

Section 1

The motivation of the project was to create a video search platform like youtube, where users can search for videos with filters of their needs, like, dislike, comment and as well as upload videos of their own. They can also see their history, get recommendations based on their action and see the performance of their videos and fight with haters in the comment chat box. Now also I visioned to separate the content of 5(Canada, France, Denmark, India, USA) different countries in different tables. So that some content is exclusive to that country(something like netflix does) and also in practical world we can put these tables on different servers across the globe. Users are allowed to view the content of other countries one at a time and need to change their settings accordingly.



Section 2

1. The source of data is this Kaggle Dataset
2. The data available is readymade
3. I majorly used 2 clean-up steps. First one being, there was no primary id in the dataset, even their were multiple copies of video_id for some unknown reason. So I had to delete repeated copies of these data points. Second one, was related to some unbalanced carriage and newline pointers in the csv files. I solved that issue using "sed" commands of linux, guided by some posts of stackoverflow.
4. So these statistics donot include for synthetically created history table which is used to maintain the activities of user across the countries and also helps in giving recommendations. Note all data base sizes are mentioned in Mb. The time taken has been mentioned for loading partially cleaned(before removing duplicates) csv files.

Name	tuples after cleanup	time in sec	size	size after cleanup
cavideos	24,427	2.17	64.1	38.4
devideos	29,627	2.67	63.0	45.36
frvideos	30,581	1.65	51.4	37.5
invideos	16,307	2.03	59.6	25.77
usvideos	6,351	2.25	62.2	9.7

Section 3

1. So each user sees 5 pages. The home page being the first, gives user to search content based on videoName, channelName or tags of the video in different countries and it can also arrange these results based on combination of different filters (like most viewed and most commented) by enabling and disabling multiple filters at the same time. Also for each video the user can watch,like,dislike and comment on the video, which affects the global stats(views,likes,..) of the video accordingly. Now the 2nd page is MyPage, here we used recursive query(with depth 3 as default) to select content based on what you liked earlier. For example if you watched a video of @MrBeast then you will get recommendations of other videos of @MrBeast as well as what @MrBeast, himself has watched(this is done recursively again). Then you have history page where you will see the latest and what type of interaction you had with any video. Now the next you have upload page to upload a video. And in the last you have login page to change your country, username(note password==username for any username), and also all the videos that you had uploaded and you can reply back to all the comments from different users here easily.

2. - I used a recursive query to get recommendations. I used index over history table(which contains all the interactions of users and is shared by all the countries) as (curr_user,videoId) since all the queries fired for this table checked only and only these 2 things. I also used an index of videoId over all the country tables since most of the qureis fired for these were checking videoId. I also added a trigger in history table, that is whenever you make any interaction with any video(watch,like,dislike,comment) your timestamp in history table gets always updated via a trigger, so that we can sort activity wrt latest timestamp. I also added a trigger for increasing comments_count attribute automatically whenever we add a comment in country tables, but removed it since it will give a useless overhead for other queries like watch,like,dislike. For virtual views in particular I didn't find any useful use so I didn't add it in the final submission. For materialized views I was also planning to add a design such that local updates of interactions with video are made in the view and the country video tables(which are larger) are updated periodically, but was not able to implement it due to shortage of time. I also added constraints while creating tables like primary key and not null for some attributes of the data.
   - So the SQL queries, excluding the one timers like creation of history table, indexes and triggers will be
     (a) search based on videoName, channelName, tags
     (b) update views, likes, dislikes in global tables
     (c) add and update comment for both viewer and owner of channel
     (d) recursive queries for recommendation
     (e) insert queries for new videos
     (f) insert or update records of user activity in history table.
     (g) search your history
     (h) search your uploaded videos
3. Now the query type in the table references to the index(a,b,..h) queries mentioned just above.

Query type	Actual Parameters	Time (in ms)
a	Channel=Marvel Entertainment	15ms
b	Watched Black Panther	12ms
c	Commented Hello	22ms
d	switched to MyPage	45ms
e	added a new video	1ms
f	watched black panther	5ms
g	switched to history page	35ms
h	switched to login page	20ms