*Explores YouTube’s business model, main users, database table design and analysis, SQLite query examples, KPIs, data security, and database architecture*

**Final Project**

Fp

ITC6000 Database Management Systems

Final Paper - YouTube

**PREPERATION:**

By: John DiSessa

For: Professor Clauson

On: May 21st ,2021

**Introduction**

*Perspective*

For my final project I demonstrated YouTube’s database management practices by using simplified query examples and table structures. YouTube is an online video platform where people can watch, create, and share videos around the world. Since YouTube was bought by Google for $1.65 billion in 2006, YouTube has exploded in popularity and service offerings. They have YouTube Kids, YouTube Music, YouTube Premium, YouTube TV, as well as capabilities to buy or rent movies and the ability to host live streams, all in addition to their primary offering. I use YouTube almost exclusively for its native video sharing capabilities, but even just their main service offering is so compelling to me that I chose it for my project. I watch videos on YouTube every day on many topics including but not limited to gaming, education, sports, and tutorials.

*Business Model*

Despite having many different services, the scope of my project is limited to only the standard YouTube offering of free videos. YouTube uses a ‘freemium’ model where their main free service is ad-based but for $11.99 a month, users can pay for YouTube Premium to have all ads removed. YouTube allows content creators to monetize their videos by allowing ads to be shown before, during, or after their videos. Even though YouTube Premium reached 30 million members in 2020, YouTube made $19.7 billion in ad revenue.

*Architecture*

YouTube is the second most frequently used social media platform in the world. On average, 2.3 billion people worldwide access YouTube at least once a month. Each day, YouTube averages 30 million daily active users. There is also a seemingly endless supply of content for users since 500 hours of videos are uploaded to YouTube every hour. Google uses their own hard drives and data centers around the world to store content and have each server ‘talk’ with other data centers and all users via the cloud. In order to maximize efficiency to ensure users have consistent and fast experiences, YouTube employs a Content Delivery Network (CDN) strategy. Essentially, videos with low views are stored at only one data center and popular videos are copied and distributed to many data centers around the world. That way users in different countries experience lower buffer times on the popular videos. Since only a small percentage of videos are that popular, this is an efficient strategy for storing and accessing trillions of gigabytes worth of videos.

**Key Personas**

*Users*

In order to maintain their status as one of the global social media leaders, YouTube focuses their efforts on the user customer experience. Users are people, like myself, who watch videos on the platform. Users also have the options to like videos, comment on videos, and subscribe to channels to easily access a channel’s library of content. Since YouTube’s primary revenue stream is from users viewing ads, it is critical they keep users happy and engaged so they keep consuming more content. One of YouTube’s algorithms is designed to recommend videos to users to keep them on the platform. The complicated algorithm looks at some key performance indicators (KPI’s) like watch-time, percent of video watched, retention rate, and engagement data (likes, comments, shares, etc.) to recommend videos based on the video length, category, channel, and other criteria. Users essentially trade off watching ads and having their data tracked and sold by Google in exchange for free content. Considering the endless amount of content available on YouTube, I personally believe it is a fair trade-off. YouTube also tracks what users do after watching a video. Do they watch another video from the same channel? Do they watch another video form the same category? Do they leave the platform (abandon rate)? All of this information factors into their algorithms.

*Creators*

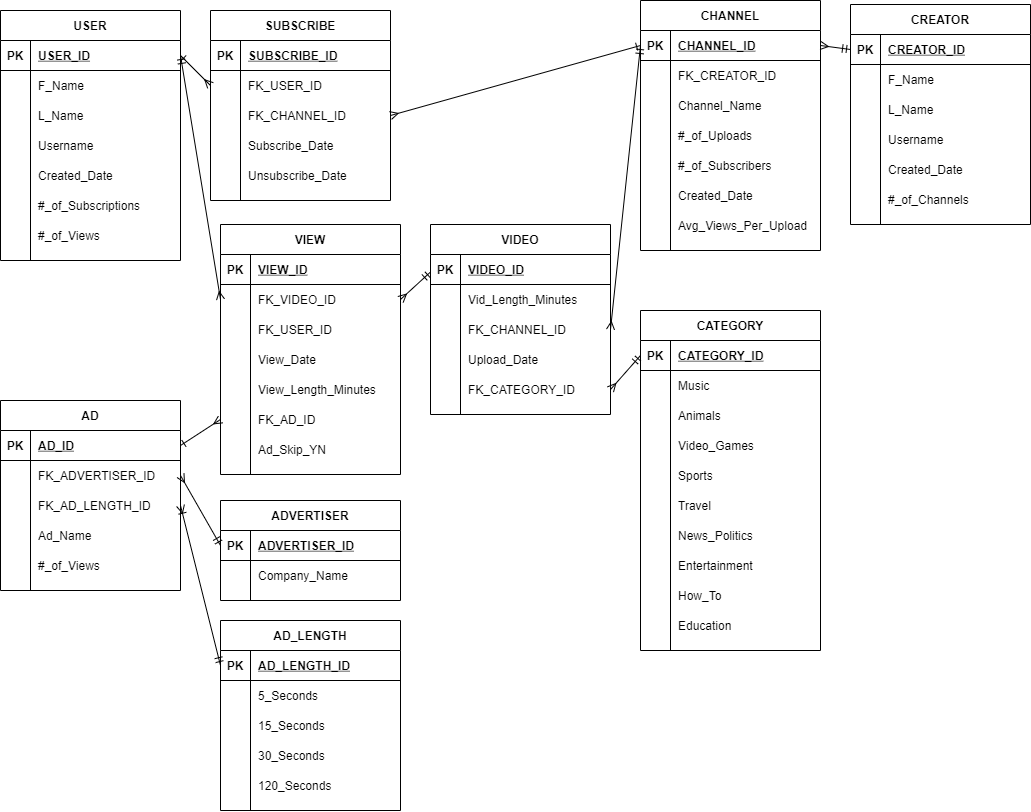
Creators are essential to YouTube because they are responsible for creating videos for users to watch. YouTube tracks creator views, views per video, subscribers gained or lost, views to engagement ratio, and other similar KPIs as they do for users in order to distinguish which creators draw more ‘eyeballs’ than others. YouTube can then show relevant ads and recommend videos to users to keep them on the ferris wheel of content consumption. If creators enable monetization on their channels, YouTube will enable ads to users watching their videos and pay creators 55% of ad revenue from advertisers (YouTube keeps 45% for themselves). Creators average between $3 - $5 per 1,000 views so, based on the number of active users on YouTube, some creators can create content as their main source of income (YouTubers).

*Advertisers*

Advertisers are the last piece to the puzzle since their expenditures make it possible for creators and YouTube to make money, thus keeping YouTube free to most users. Advertisers create videos to be shown to users via videos. They pay YouTube a few cents per view and can set criteria to target specific users. Advertisers tell YouTube their budget, the type of user they want to target, the geographical location they want to target, and other information. Another YouTube algorithm computes it, compares it to other advertiser’s budgets, and then shows ads according to their revenue optimization models. Since it is very hard to get users to leave the videos they chose to watch for an advertisement, it is critical YouTube tracks click-through rate so YouTube and advertisers know which ads get positive return on investment. They also track duration of ad watched and whether the ad was skipped or not.

**Entity Relationship Diagram & Tables**

I created the following Entity Relationship Diagram to show the basics of how YouTube stores data and how different data tables interact.



*User*

The user table holds information for each user such as their name, username, date their account was created, as well as an automatic aggregate of their total views and number of subscriptions. Each row in the user table corresponds to a unique user and a new row gets inserted for each new account created.

*Creator*

The creator table holds similar information to the user table except this is for people who want to create content on YouTube. In reality only one account is needed since YouTube allows each account to post or view videos. For the sake of simplicity, I treated them separately. Each row in the creator table corresponds to a unique creator account and a new row gets inserted for each new account created.

*Channel*

A creator can create one or many channels through which they post videos. Each channel shows who the channel creator is, the name of the channel, the date it was created, an aggregate of the total uploaded videos per channel, the number of subscribers each channel has, and the average views per video. Every time a creator creates a channel, a new row gets inserted into the channel table. It is important to distinguish between a creator and a channel. It is common for creators to have multiple channels in order to cater to different audiences or post different types of content. YouTube would not want individuals having multiple creator accounts because it creates duplicate information about the same individuals.

*Video*

Each video uploaded to YouTube is stored here with a unique primary key. Currently I track which channel uploaded the video, when it was uploaded, what category it is, and how long the video is. Each video is assigned a category from the category table. Some examples include sports, animals, and music. I only allow for one category to be assigned to a video, but YouTube in real-life allows for multiple categories to be selected.

*View*

The view table is perhaps the most important table since YouTube’s revenue is directly tied to views. Every time a user watches a video, a new row is created in this table with a time stamp, how long the video was viewed for, which ad was shown, and if the ad was skipped. For every view, there has to be a user and a video. This table stores every combination of user\_id and video\_id to produce a unique view. In reality, YouTube also uses the date a video was watched as another unique identifier because one user could watch the same video multiple times. They also have logic where if a video is only watched for a few seconds, the view doesn’t count in order to avoid robots inflating view counts. However, for the scope of this project I ignored these two facts.

*Subscribe*

The subscribe table acts similarly to the view table in that it stores every combination of user\_id and channel\_id with a new row inserted every time a user subscribes to a channel. I also store the date the user subscribed and the optional date they unsubscribed. YouTube would have analytics to see which types of users are more likely to subscribe to a certain type of channel as well as subscribing trends, however I kept this table simple.

*Ad*

The ad table has a unique row for each ad created by all of the companies trying to advertise on YouTube. Each ad has an advertiser\_id which links to the advertiser table. The advertiser table assigns a unique id for each company. This allows for one company to have multiple ads. The ad table also assigns a value based on the ad\_length\_id from the ad\_length table. YouTube does not allow ads to be any duration. YouTube determined different ad length categories that advertisers must use in order to fit into their algorithm. Based on their years of data analysis, they know that 15 second ads are less engaging than 5 second ads, and 30 second ads are less engaging than 15 seconds ads. YouTube also needs ads to be consistent lengths so they can properly incorporate the ‘skip’ feature. It would make for a horrible user experience if 5 second ads could be skipped but 30 second ads could not be.

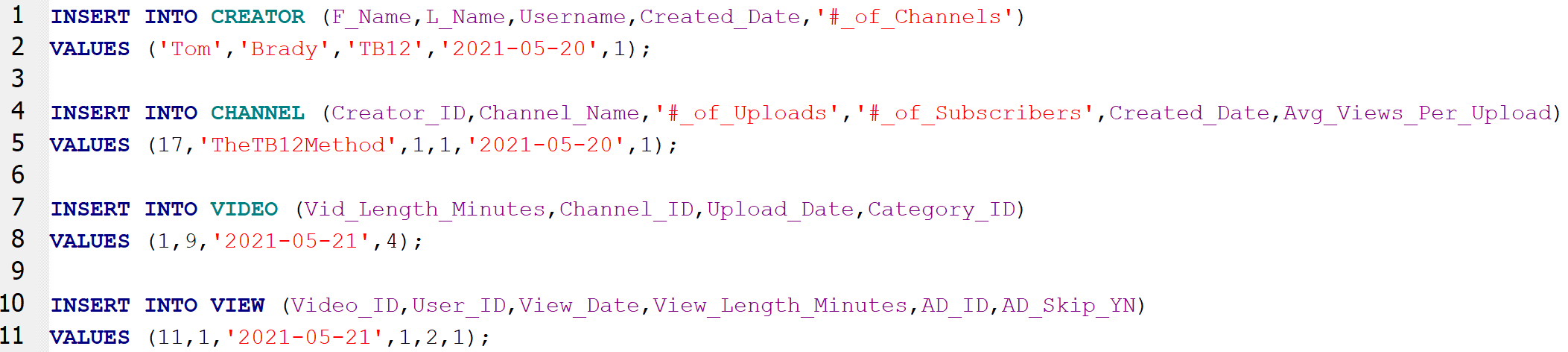
*Business Rules*

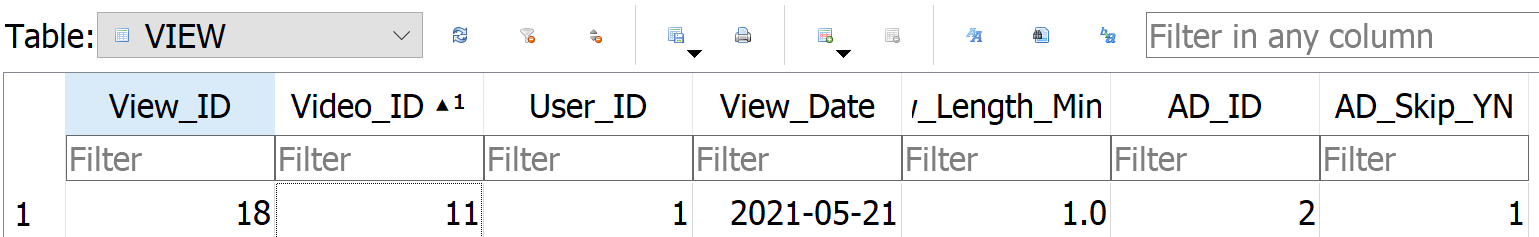
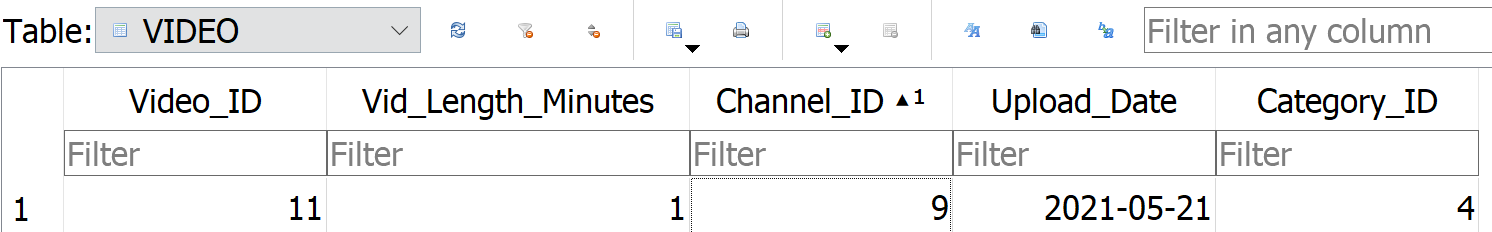
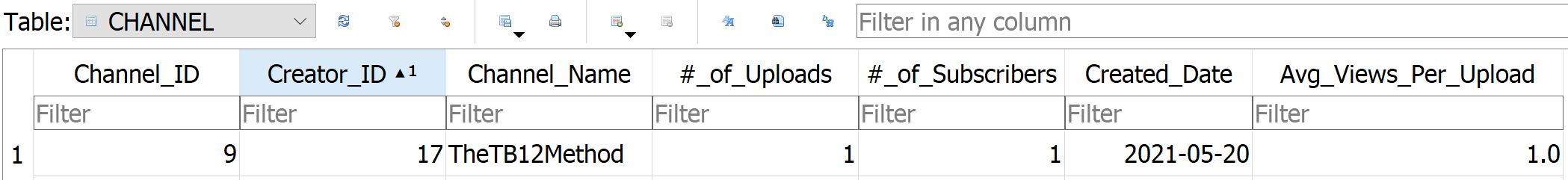
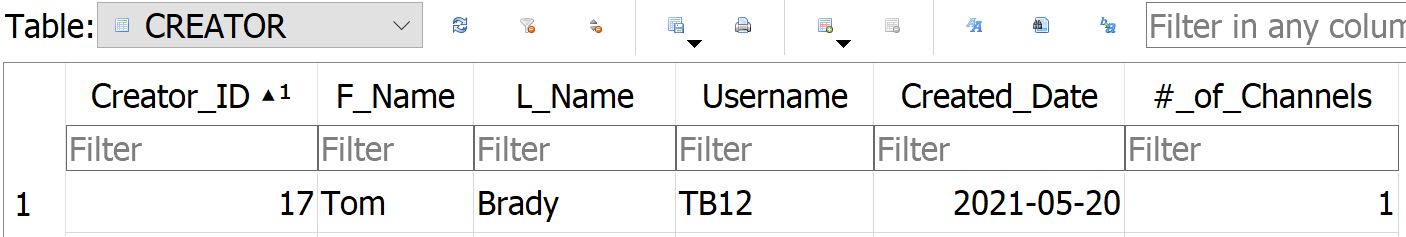
* One creator can have many channels
* One channel can have many videos
* One category can be assigned to many videos
* One video can have many views
* One user can have many views
* One user can subscribe to many channels
* One channel can have many subscribers
* One ad can be shown on many views
* One advertiser can have many ads
* One ad length can be assigned to many ads

**Database Implementation**

*Creator Lifecycle*

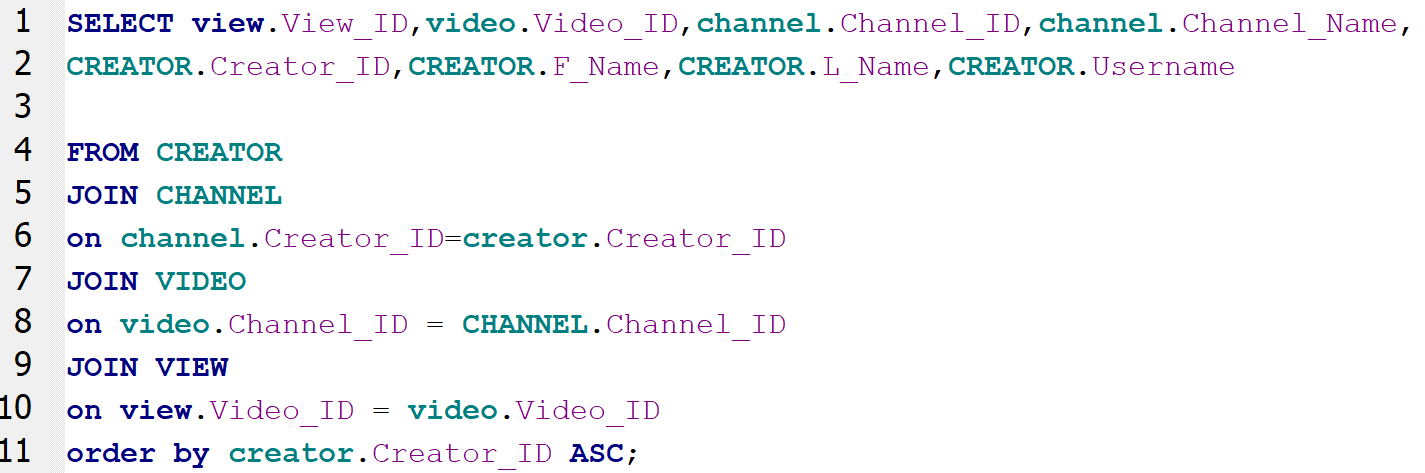
The following queries take us through the data upload and storage process when a creator creates a new account, channel, video, and views. In this example, Tom Brady created an account on 5/20/2021 with a username of ‘TB12’. A new row was inserted into the creator table. Tom then created a channel called ‘TheTB12Method’ also on 5/20/2021. From the creator table he was assigned a unique primary key (Creator\_ID = 17) so this becomes the foreign key on the channel table in order to link them together. The next day, ‘TheTB12Method’ uploaded a 1 minute long video under the sports category (Category\_id = 4) so a new row was inserted into the video table. Later that day, a user (User\_id = 1) watched Tom’s video (Video\_id = 11) for 1 minute, was shown an ad (Ad\_id = 2) and skipped the ad (Ad\_skip\_YN = 1).

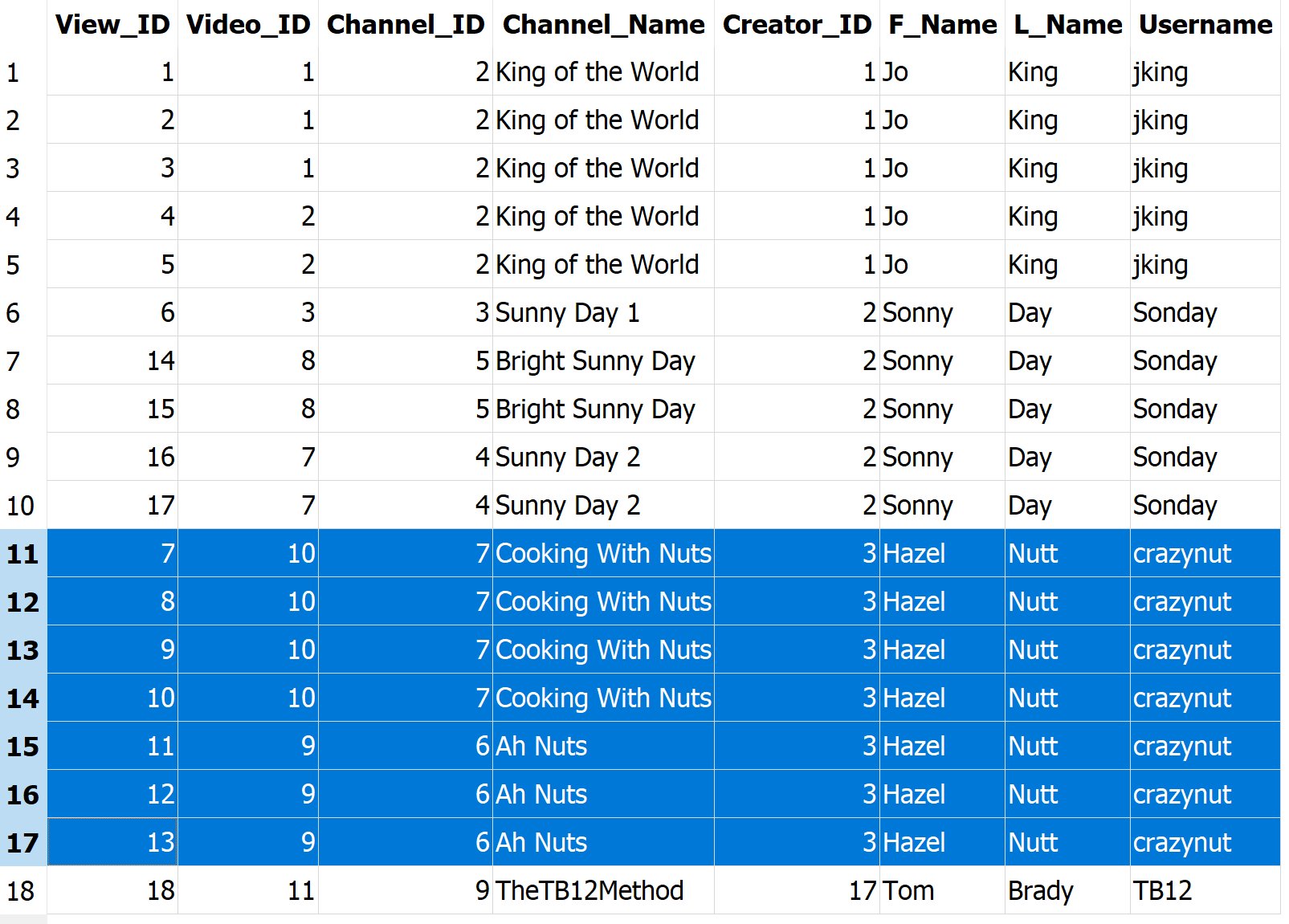




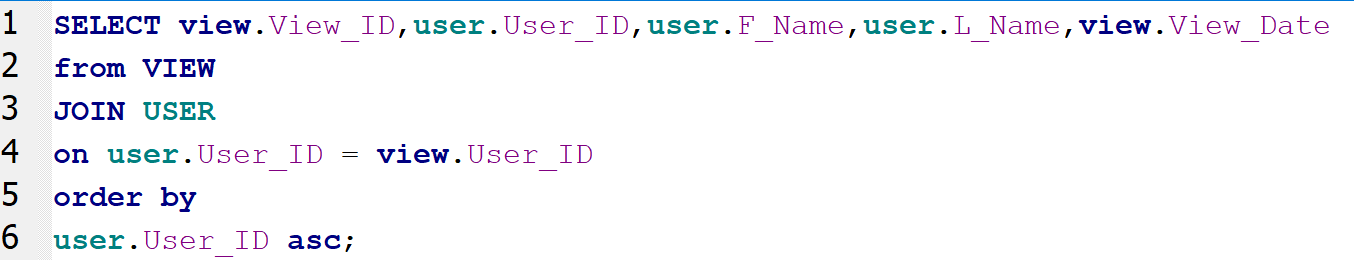
*Total Views per Creator*

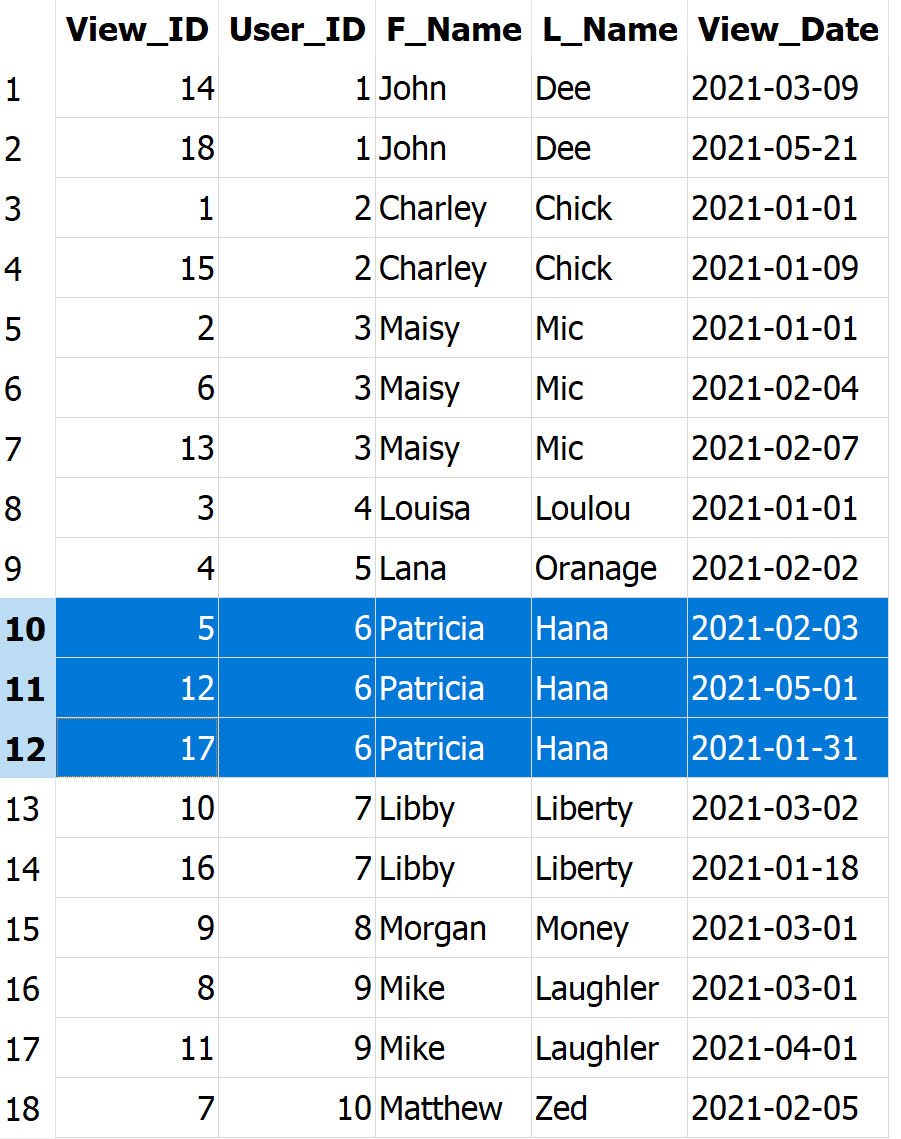
Since YouTube and creator revenues are tied to views, it is critical YouTube knows which content creators are more valuable to the platform than others so that they can compensate their creators fairly. The following query shows the total views grouped by creator. The query connects the creator table to the channel table to the video table to the view table. As shown in the results below. Hazel Nutt has the most views of any creator with 7 across her 2 channels. Hazel Nutt should earn a little more than Sonny Day and Jo King, but a lot more than Tom Brady since he only has 1 view.





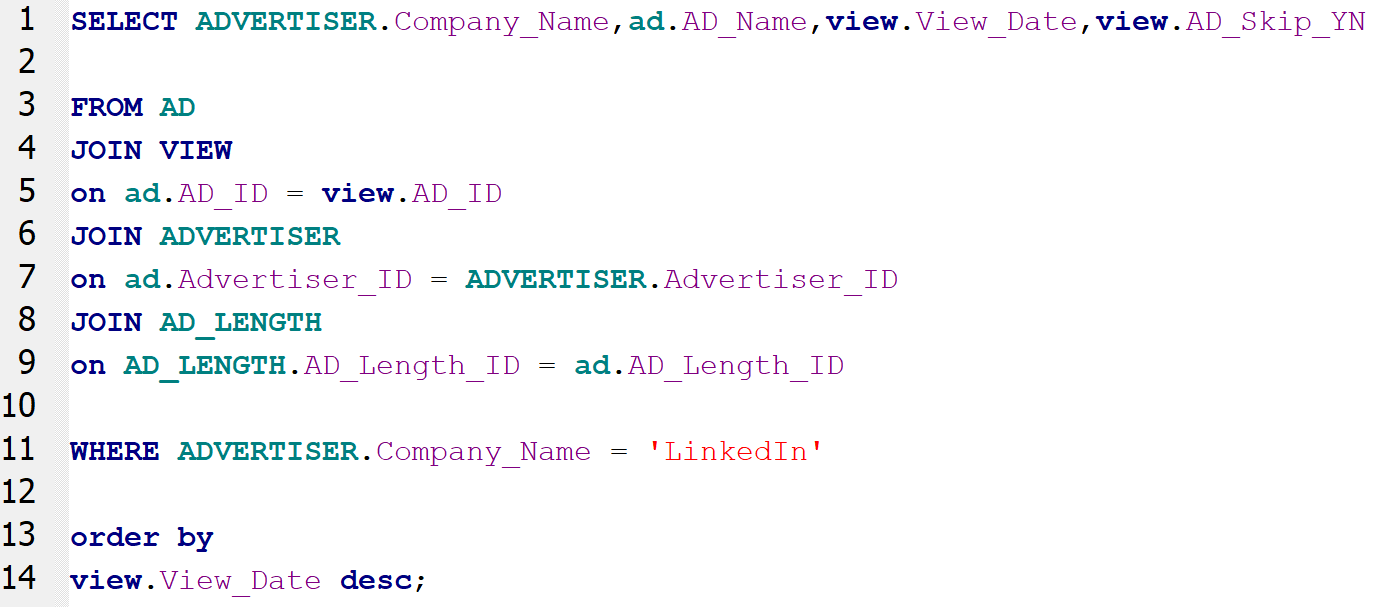
*Users with the Most Views*

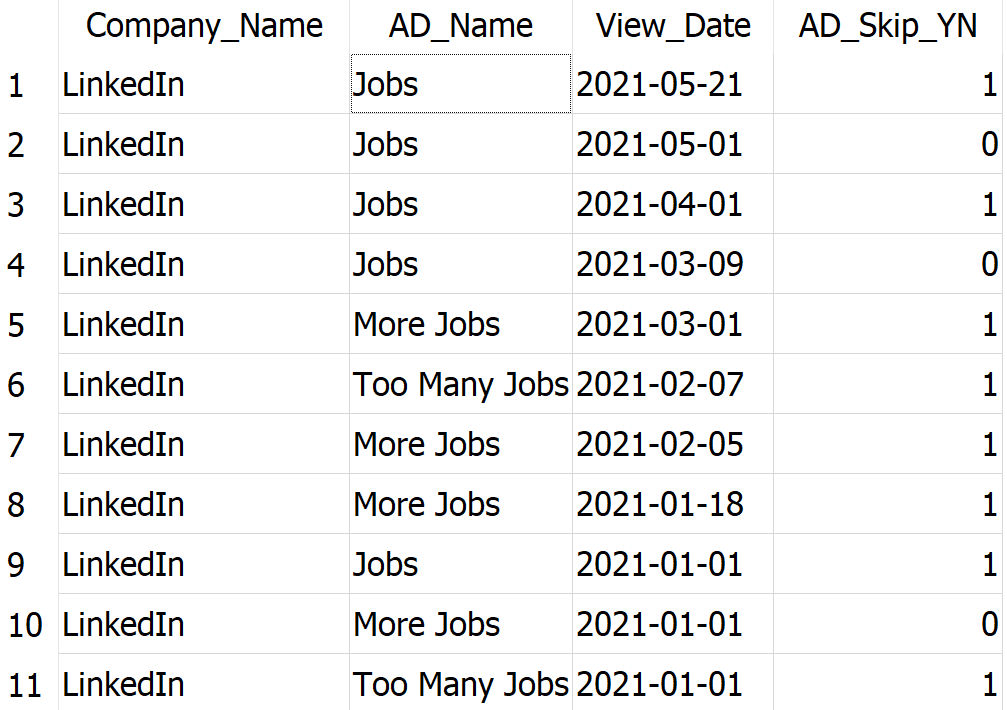
Knowing which users have the most views has important business implications as well. By knowing which users have the most views, YouTube can study their watch patterns and have the algorithms recommend videos to other users so their view counts can increase as well. As you can see from the query below, Patricia Hana has the most views with 3. This query joins the view and user tables. 



*LinkedIn Ad Profitability*

Since ad revenue is YouTube’s largest source of income, it is necessary they know which companies produce engaging ads and which don’t. For this example, LinkedIn is one of YouTube’s biggest advertisers. However, YouTube needs to know how much value they add to their business by seeing if their ads are resonating with users. Aside from click through rate, my database is capable of tracking if users skipped or did not skip an ad from LinkedIn. From the results below, 8 ads out of the 11 shown were skipped. YouTube can then show LinkedIn this data when they try to negotiate for more favorable terms.





**Data Security**

*Security Concerns*

YouTube tracks and stores a vast amount of data on its users. Some of which would be classified as Personally Identifiable Information (PII) such as name, email, street address, phone number, and credit card information. A lot of the data they track would not classify as PII such as viewing habits, region, likes, comments, subscription, etc. However, that information is still valuable to hackers. Since YouTube uses that data to justify charging advertisers a certain amount per view, there could be malicious groups that could sell our data to other companies if they got access. Even if no PII was hacked, viewing habits are exclusive data points that YouTube needs to protect for their core business. PII hacks would jeopardize their credibility with users and potentially lead to less active users (and thus less ad views). Even though I have all user information stored in the user table, YouTube stores that information across multiple tables. That way if one table is hacked the hackers do not get access to all the data at once. They would need to hack many tables which is far less likely. YouTube asks users to take on some of the responsibility in data privacy by recommending they have strong passwords, enable 2-step verification, and are encouraged to report any malicious activity. YouTube is also continuously improving their own data security. In 2017, YouTube announced they created a new security chip called Titan which prevents hacks at each data center. Any time information is accessed, it automatically compares it to previous iterations. If it matches, it allows the data to be accessed. Hackers would then need to, not just hack current data, but all previous iterations of the data they want. While it is not 100% secure it is still a massive step in prevent cyber-attacks.

**Conclusion**

The biggest takeaway I personally had when doing this project was it really showed me how much value proper data storage adds to a business. When you think of YouTube analytics you think of the algorithms to suggest videos and optimize ads, which is critically important. But that can’t be done unless the data is stored properly and efficiently. I am also extremely grateful to have learned and practice writing queries. In my first job out of college I was thrust in a role where I had to create queries despite having no prior experience or training. I became really frustrated and it put a sour taste in my mouth about SQL. I am happy to now understand queries and see the value for business data storage. I am hoping to use this knowledge to market myself to employers upon completion of my degree program.

If I were to take additional steps in this project, the next step would be to add tables to store likes and comments. I would create intermediary tables, like I did with the subscribe table, to track every like and comment. I would also expand the ad and view tables to include options to have the ad placed at the beginning, middle, or end of a video. Right now, I do not make a distinction. After that, I could have queries scheduled to run every so often to constantly update total view counts, total subscriber counts, etc

**Citations**

“All the YouTube Services: YouTube Music, YouTube Premium, and More.” *Android Authority*, 10 May 2021, www.androidauthority.com/youtube-service-984598/.

Luckerson, Victor. “A Decade Ago, Google Bought YouTube - and It Was the Best Tech Deal Ever.” *The Ringer*, The Ringer, 10 Oct. 2016, www.theringer.com/2016/10/10/16042354/google-youtube-acquisition-10-years-tech-deals-69fdbe1c8a06.

“The Business Model of YouTube.” *Business Models Inc.*, 15 Dec. 2020, www.businessmodelsinc.com/business-model-youtube/.

“YouTube Revenue and Usage Statistics (2021).” *Business of Apps*, 14 May 2021, www.businessofapps.com/data/youtube-statistics/.

“How Does YouTube Store And Analyze Such Huge Amount of Data?” *GeeksforGeeks*, 10 Dec. 2019, www.geeksforgeeks.org/how-does-youtube-store-and-analyze-such-huge-amount-of-data/.

“The 17 Essential YouTube KPIs for Measuring & Growing Your Channel: Databox Blog.” *Databox*, 8 Apr. 2021, databox.com/youtube-kpis.

“Guide to Identifying Personally Identifiable Information (PII).” *Guide to Identifying Personally Identifiable Information (PII) | Information Technology | University of Pittsburgh*, 16 Feb. 2021, www.technology.pitt.edu/help-desk/how-to-documents/guide-identifying-personally-identifiable-information-pii#:~:text=Personal%20identification%20numbers%3A%20social%20security,Personal%20telephone%20numbers.

12, Maria Korolov | Nov. “Google's OpenTitan Chip Could Make Data Centers More Secure... Someday.” *Data Center Knowledge*, 12 Nov. 2019, www.datacenterknowledge.com/google-alphabet/google-s-opentitan-chip-could-make-data-centers-more-secure-someday.