Ali Shakir, Krish Patel, Shivam Patel

CNIT 37200

Final Report

Background

Our main goal is to really dive into YouTube data to make our channel better and attract more viewers. Since the online world is getting more crowded, using data to make smart decisions is key, especially on a big platform like YouTube.

To get started, we're looking at three important sets of data. At the top of the list is the "YouTube Channel and Influencer Analysis" (Data 1). This set will show us what makes a YouTube channel popular and what types of videos people like to watch. By understanding this, we can create videos that our viewers will love and want to watch again and again.

Using this information, we can plan our videos more wisely. We'll know what our viewers like and can create more of that type of content. This way, our current viewers will stick around, and we'll also bring in new ones. By combining our creativity with what the data tells us, we can make sure our videos reach as many people as possible.

The team has decided to work with a dataset we're calling Data 1. After checking out the information inside, we noticed that it could be split neatly into four main parts. So, we're going to organize it into four tables: one for videos, one for creators, one for channels, and one for interactions (like comments or likes).

When we looked closely at the dataset, it made sense to split the information this way because each category had its own set of details. For example, the video table would have things like video titles and lengths, while the creator table would have names and maybe contact info. Now,

to make sure our database runs smoothly and everything is connected properly, we needed something unique to link everything together. We chose the Creator's name as this special link because no two creators can have the exact same name. This unique name will act like a passport, helping us connect details across the different tables. By using the Creator's name in this way, it's easier for us to find and organize all the related information when we need it.

Database Description

For Data 1, the information that we would need would be split into 4 tables. The table names would be Video, Creator, Channel, and Interaction.

Video table:

- Link(string)
- VideoViews(integer)
- Title(string)
- Language(string)
- Quality(Integer)

Creator Table:

- Creator Name(string)
- Creator Gender(string)
- Link(string)

Channel Table:

- SubCount(integer)
- ChannelViews(integer)

- NumVideos(integer)
- NumPlaylist(integer)
- Link(string)

Interaction Table:

- Numlikes(integer)
- NumComments(integer)
- CommunityEngagement(integer)
- Link(string)

Solutions

1. Identifying Trending Video Languages:

"Write a query to determine which language has the highest average video views, indicating potentially trending content languages."

```
SELECT Language, AVG(VideoViews) AS AverageViews
FROM Video
GROUP BY Language
ORDER BY AverageViews DESC;
```

Help us find the average number of video views for each language in the so we can see which language has the highest average video views. The language with the average number of video views was English.

2. Analyzing Video Quality Impact on Engagement:

"How can you analyze if there's a correlation between video quality and community engagement (likes and comments)?"

```
SELECT Quality, AVG(Numlikes + NumComments) AS AverageEngagement
FROM Video v
JOIN Interaction i ON v.Link = i.Link -- Assuming Link as the common identifier
GROUP BY Quality;
```

Since a significant correlation was found, it suggests that there may be a relationship between video quality and community engagement. This information can be valuable for content creators and platform administrators as they might change their strategies for posting

3. Evaluating Creator Gender Diversity in Popular Content:

"Create a query to evaluate the gender diversity of creators among the top 10% most viewed videos."

-- This query requires a modification in the schema to link creators with videos.

The gender diversity of creators among the the top 10% most viewed videos was male.

4. Strategizing Channel Growth:

"How would you identify channels that have a high number of subscribers but relatively low video views, suggesting potential areas for content improvement?"

```
SELECT
    c.Link,
    c.SubCount,
    SUM(v.VideoViews) AS TotalVideoViews
FROM
    Channel c
JOIN
    Video v ON c.Link = v.Link
GROUP BY
    c.Link, c.SubCount
HAVING
    c.SubCount > 500 AND SUM(v.VideoViews) < 5000000;
```

The most common video language among the top 20% most-liked was English. Spanish is also fairly high.

5. Assessing Community Engagement Trends:

"Write a SQL query to find out if there's a significant difference in community engagement (likes + comments) between videos with different quality levels."

```
SELECT Quality, AVG(Numlikes + NumComments) AS AverageEngagement
FROM Video v
JOIN Interaction i ON v.Link = i.Link
GROUP BY Quality;
```

6. Optimizing Content for Viewer Preferences:

"Can you determine the most common video language among the top 20% most-liked videos, to understand viewer preferences?"

-- This query requires a ranking or percentile function that is not directly supported with the given table structure.

7. Channel Performance Analysis:

"How would you compare the average number of views per video against the total subscriber count for each channel, to assess overall channel performance?"

```
SELECT
    c.Link,
    c.SubCount,
    AVG(v.VideoViews) AS AverageViewsPerVideo
FROM
    Channel c
JOIN
    Video v ON c.Link = v.Link
GROUP BY
    c.Link, c.SubCount;
```

This question gave us an insight to potentially see if any channels were outperforming the number of views they get based off a lower subscriber count.

8. Creator Impact on Video Popularity:

"Create a query to find out which creators' videos have, on average, the highest number of views, indicating their influence on content popularity."

-- This query requires linking creators with videos, which is not possible with the current table structure.

9. Understanding Audience Interaction Patterns:

"Write a SQL query to analyze the ratio of likes to comments across all videos, to understand audience interaction patterns."

SELECT AVG(Numlikes) AS AverageLikes, AVG(NumComments) AS AverageComments
FROM Interaction;

We were able to see that higher amounts of likes leads to a larger amount of comments, which was expected.

10. Evaluating Channel Engagement Efficiency:

"How can you assess channels based on their 'engagement efficiency,' calculated as total community engagement divided by the number of videos, to identify which channels are most effectively engaging their audience?"

```
SELECT
    c.Link,
    (SUM(i.Numlikes + i.NumComments + i.CommunityEngagement) / c.NumVideos) AS
EngagementEfficiency
FROM
    Channel c
JOIN
    Interaction i ON c.Link = i.Link
GROUP BY
    c.Link, c.NumVideos;
```

Team:

Ali Shakir- Ali talked about the different datasets and the insights they may provide. Discussed the importance of using data for content improvement and suggested questions related to video engagement and viewership trends. created the git repository, uploaded the data into the tables, created the script files, and answered questions 1,2, and 3. Ali was also in charge of gathering all information needed for the final report and organized the files into their correct folders.

Krish Patel- Krish was in charge of outlining the tables that we plan to work with and describing the column names, the data types, and the constraints that they may have. Outlined the questions and explained how they relate to class knowledge, including data analysis techniques and their relevance to content creators. tested all the code again in order to make sure everything worked. He also answered questions 4,5,6,7.

Shivam Patel- Shivam was in charge of explaining how we would design the database to store all of the data we have. Provided insights into the potential uses of the data and contributed questions related to language analysis, subtitles, and community engagement. created the questions, answered questions 8, 9, 10.