A link to your GitHub repository.
https://github.com/aliscores/Group8.git

-In your repository, there should be

- A script for importing data into an empty database.
- The project description from Milestone 1 and the question list from Milestone 2

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;

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;

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.

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;
```

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;
```

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;
```

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;
```

- The solutions you have implemented,labeled by the corresponding questions.

-Describe your teamwork: how did you come up with the 10 questions, list the contribution of each team member.

When coming up with the 10 questions for the assignment, we were thinking about ways that we could see what the best creators are doing on their channels. We wanted to look at subscriber counts, views, likes, and comments.

Ali created the git repository, uploaded the data into the tables, created the script files, and answered questions 1,2, and 3.
Krish tested all code again in order to make sure everything work. He also answered questions 4,5,6,7.
Shivam created the questions, answered questions 8, 9, 10.