**QUERIES FOR THE DATA HARVESTING PROJECT – GUVI**

IA logo for a company

Description automatically generated

1.What are the names of all the videos and their corresponding channel

Ans :-

SELECT v.video\_name, c.channel\_name

FROM video v

JOIN channel c ON v.channel\_id = c.channel\_id;

Explanation :

This query selects the video name from the video table and the channel name from the channel table, joining them on the channel\_id column to ensure that each video is associated with its corresponding channel. This will give you the names of all videos along with their corresponding channel names.

2. If playlist table was not barred we could have used :

SELECT v.video\_name, c.channel\_name, p.channel\_name

FROM video v, channel c, playlist p

JOIN playlist p ON v.playlist\_id = p.playlist\_id

JOIN channel c ON p.channel\_id = c.channel\_id;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2.Which channels have the most no of videos and how many videos do they have.

Ans:-

SELECT c.channel\_id, c.channel\_name, COUNT(v.video\_id) AS num\_videos

FROM channel c

JOIN video v ON c.channel\_id = v.channel\_id

GROUP BY c.channel\_id, c.channel\_name

ORDER BY num\_videos DESC;

EXPLAINATION:

This query retrieves the channel ID and name from the channel table, counts the number of videos associated with each channel using the video table, and groups the results by channel. It then orders the result set by the number of videos in descending order, so you'll see the channels with the most videos at the top.

3.What are the top 10 most viewed videos and their respective channels

Ans :

SELECT v.video\_name, c.channel\_name, v.view\_count

FROM video v

JOIN channel c ON v.channel\_id = c.channel\_id

ORDER BY v.view\_count DESC

LIMIT 10;

EXPLAINATION:

This query selects the video name from the video table, the channel name from the channel table, and the view count from the video table. It joins the two tables on the channel\_id column to ensure that each video is associated with its corresponding channel. The results are then ordered by view count in descending order and limited to the top 10 rows, giving you the top 10 most viewed videos and their respective channels.

4. How many comments are made on each video and their corresponding video names.

Ans :

SELECT v.video\_name, COUNT(c.comment\_id) AS num\_comments

FROM video v

JOIN comment c ON v.video\_id = c.video\_id

GROUP BY v.video\_name;

EXPLAINATION:

This query selects the video name from the video table and counts the number of comments associated with each video using the comment table. It joins the two tables on the video\_id column to ensure that each comment is associated with its corresponding video. The results are then grouped by video name, giving the number of comments made on each video along with their corresponding video names.

5.Which videos have the highest number of likes and what are their corresponding vide names.

Ans :

SELECT v.video\_name, v.like\_count

FROM video v

ORDER BY v.like\_count DESC

LIMIT 10;

**EXPLAINATION:**

This query selects the video name and like count from the video table. It then orders the results by like count in descending order and limits the output to the top 10 rows. This will give the videos with the highest number of likes and their corresponding names.

6.What is the total number of likes and dislikes for each video , and what are their corresponding video names

Ans :

SELECT

v.video\_name,

SUM(v.like\_count) AS total\_likes,

SUM(v.dislike\_count) AS total\_dislikes

FROM

video v

GROUP BY

v.video\_name;

EXPLAINATION:

This query selects the video name from the video table and sums up the like\_count and dislike\_count for each video. It then groups the results by video name. This will give you the total number of likes and dislikes for each video along with their corresponding video names.

7. What are the total number of views for each channel, and what are their corresponding channel names.

Ans:

SELECT c.channel\_name, SUM(v.view\_count) AS total\_views

FROM channel c

JOIN video v ON c.channel\_id = v.channel\_id

GROUP BY c.channel\_name;

**EXPLAINATION:**

This query selects the channel name from the channel table and sums up the view\_count for each channel's videos using the video table. It then groups the results by channel name. This will give you the total number of views for each channel along with their corresponding channel names.

8.What are the names of all the channels that have published videos in the year 2022

Ans :

SELECT DISTINCT c.channel\_name

FROM channel c

JOIN video v ON c.channel\_id = v.channel\_id

WHERE YEAR(v.published\_date) = 2022;

**EXPLAINATION:**

This query selects distinct channel names from the channel table that have published videos in the year 2022. It joins the channel and video tables on the channel\_id column, filters the videos based on the published\_date falling within the year 2022, and then retrieves the distinct channel names.

9. What is the average duration of all the videos in each channel, and what are their corresponding channel names.

Ans :

SELECT c.channel\_name, AVG(TIME\_TO\_SEC(v.duration)) AS avg\_duration\_seconds

FROM channel c

JOIN video v ON c.channel\_id = v.channel\_id

GROUP BY c.channel\_name;

**EXPLAINATION:**

This query selects the channel name from the channel table and calculates the average duration of all videos in each channel using the video table. The duration is converted to seconds using the TIME\_TO\_SEC function before calculating the average. The results are then grouped by channel name. This will give you the average duration of all the videos in each channel along with their corresponding channel names.

10.Which videos have the highest number of comments and what are their

Corresponding channel names.

Ans :

SELECT v.video\_name, c.channel\_name, COUNT(co.comment\_id) AS num\_comments

FROM video v

JOIN channel c ON v.channel\_id = c.channel\_id

JOIN comment co ON v.video\_id = co.video\_id

GROUP BY v.video\_name, c.channel\_name

ORDER BY num\_comments DESC

LIMIT 10;

**EXPLAINATION:**

This query selects the video name from the video table, the channel name from the channel table, and counts the number of comments associated with each video using the comment table. It then joins the three tables together on their respective IDs and groups the results by video name and channel name. Finally, it orders the results by the number of comments in descending order and limits the output to the top 10 rows. This will give you the videos with the highest number of comments and their corresponding channel names.