Project Cont'd

Consider an online video-sharing platform like YouTube which hosts tens of thousands of channels and crores of users.

You have to analyse the data and provide meaningful insights on the type of content that drives engagement, users growth, and many more to all the stakeholders. Let's roll our sleeves up for an insightful analysis!

Database

The sample database consists of tables that store the information of users, channels, videos, genres and likes/dislikes.

Note:

channel_user table

channel_id	user_id	subscribed_datetime
100	1	2020-12-10 10:30:45
100	7	2020-10-10 11:30:45
...	...	...

channel_user table stores the data of the channel_ids and their subscribers' user_ids.

First row in the table represents that the user with user_id = 1 is subscribed to the channel with channel_id = 100 at

2020-12-10 10:30:45

user_likes table

user_id	video_id	reaction_type	reacted_at
1	10	LIKE	2020-12-10 10:30:45
7	10	DISLIKE	2020-10-10 11:30:45
...	...	...	...

Similarly,

user_likes table stores the data of video_id and the user_ids who reacted to the video.

video_genre table

video_id	genre_id
10	201
10	202
...	...

Similarly,

`video_genre` table stores the data of video_id and the ids of the genres that the corresponding video belongs to.

Let's dive in to analyze the in and outs of each part of the data. Here we go!

QUESTIONS

1. Get the total number of users in the platform as `users_count`.

   Expected Output Format

users_count
...

SELECT video_id, name, (no_of_views / 1000.0) AS no_of_views_in_thousands FROM video ORDER BY no_of_views_in_thousands DESC, name ASC

2. Get the total number of *distinct* countries where the users are located. Country of the user is present in the `user` table.

   Expected Output Format

countries_count
...

SELECT count(DISTINCT country) AS countries_count FROM user

3. Get the number of videos uploaded by each channel.

   Expected Output Format

channel_id	videos_count
...	...

SELECT channel_id, count(name) AS videos_count FROM video GROUP BY channel_id

4. Get the ids of all the channels that have uploaded at least 50 videos.

   Note:

   - Sort the output in the ascending order of the `channel_id`

Expected Output Format

channel_id
...

SELECT channel_id FROM video GROUP BY channel_id HAVING count(name) >= 50

5. For all the videos, represent the number of views in multiples of thousands.

   For example, if the number of views of a video is 17,200, it is represented as 17.2 in the output.

   Note:

   - Sort the output in the descending order of `no_of_views_in_thousands`, and then in the alphabetical order of `name`

   Expected Output Format

video_id	name	no_of_views_in_thousands
...	...	...

SELECT video_id, name, (no_of_views / 1000.0) AS no_of_views_in_thousands FROM video ORDER BY no_of_views_in_thousands DESC, name ASC

6. Find the sum of durations of the videos published by each channel in hours.

   Note:

   - The output must contain the duration as number of hours.
   - Sort the output in the descending order of the `no_of_hours`

   Expected Output Format

channel_id	no_of_hours
...	...

SELECT channel_id, sum(duration_in_secs / 3600.0) AS no_of_hours FROM video GROUP BY channel_id ORDER BY no_of_hours DESC

7. Categorise the performance of all the videos released by the "Motivation grid" Channel (id = 350).

   Performance of a video is measured based on the number of views of the video.

   Categorization

no_of_views	category
<= 10000	poor
10000 < views <= 100000 and	average

no_of_views	category
> 100000	good

Note

Sort the output in the ascending order of

published_datetime

Expected Output Format

name	no_of_views	category
...	...	...

SELECT name, no_of_views, CASE WHEN no_of_views <= 10000 THEN 'poor' WHEN no_of_views > 10000 AND no_of_views <= 100000 THEN 'average' ELSE 'good' END AS category FROM video WHERE channel_id = 350 ORDER BY published_datetime ASC

8. Get the number of videos released in each year.

Note:

- For this question, convert the year in string datatype to INT datatype.
- Sort the output in the ascending order of year

Expected Output Format

year	no_of_videos
...	...

SELECT cast(strftime("%Y", published_datetime) AS integer) AS year, count(*) AS no_of_videos FROM video GROUP BY year ORDER BY year

9. For Marvel channel (id = 351), get the number of subscribers added in each month in the year 2020.

Note:

- You can find the subscribed date of a user for a channel in the channel_user table.
- For this question, convert the month_of_year in string datatype to INT datatype.
- Sort the output in the ascending order of the month.

Expected Output Format

month_of_year	no_of_subscribers
...	...

SELECT cast(strftime("%m", subscribed_datetime) AS integer) AS month_of_year, count(*) AS no_of_subscribers FROM channel_user WHERE cast(strftime("%Y", subscribed_datetime) AS integer) = 2020 AND channel_id = 351 GROUP BY month_of_year ORDER BY month_of_year ASC

10. Get the number of reactions (likes/dislikes) generated in each hour of the day in the year 2020.

Note:

- For this question, convert the `hour_of_day` in string datatype to INT datatype.
- Sort the output in the ascending order of `hour_of_day`

Expected Output Format

hour_of_day	no_of_reactions
0	500
1	2450
..	..
..	..
23	400

```
SELECT cast(strftime("%H", reacted_at) AS integer) AS hour_of_day, count(*) AS no_of_reactions FROM user_likes WHERE cast(strftime("%Y", reacted_at) AS integer) = 2020 GROUP BY hour_of_day ORDER BY hour_of_day ASC
```

11. Get all the channel_ids that uploaded at least one video in "AI/ML" or "Robotics" technologies between 2018 and 2021.

Note:

- Consider all the videos that have any of the technologies mentioned above in their `name`
- Sort the output in the ascending order of `channel_id`

Expected Output Format

channel_id
...

```
SELECT DISTINCT channel_id FROM video WHERE (name LIKE '%AI/ML%' OR name LIKE '%Robotics%') AND (cast(strftime("%Y", published_datetime) AS integer) BETWEEN 2018 AND 2021) ORDER BY channel_id ASC;
```

12. Get all the channel_ids that uploaded at least 20 videos in "AI/ML", "Cyber Security", "Data Science" or "Robotics" technologies between 2018 and 2021.

Example: If a channel publishes 5 videos in AI/ML, 10 videos in Cyber Security and 5 videos in Data Science, consider the channel.

Note:

- Consider all the videos that have any of the technologies mentioned above in their `name`
- Sort the output in the ascending order of channel_id.

Expected Output Format

channel_id

...

SELECT channel_id FROM video WHERE (name LIKE '%AI/ML%' OR name LIKE '%Robotics%' OR name LIKE '%Data Science%' OR name LIKE '%Cyber Security%') AND (cast(strftime("%Y", published_datetime) AS integer) BETWEEN 2018 AND 2021) GROUP BY channel_id HAVING count(name) >= 20 ORDER BY channel_id AS