User

user_id	name	gender	age	country	premium_membership
2000	John White	M	63	AUSTRALIA	1
2001	John Andrews	M	67	AUSTRALIA	1
2002	April Robinson	F	60	SRILANKA	1
2003	Kathy Ryan	F	81	SRILANKA	0

Channel

channel_id	name	owner_id	created_datetime
350	Motivation grid	1011	2014-10-05 17:32
351	Marvel	1011	2014-10-05 17:32

Video

video_id	name	duration_in_secs	published_datetime	no_of_views	channel_id
1000	Getting My Driver's License | Lele Pons	3652	2011-12-05 19:00	10619	367
1001	Apple iPhone X Review: The Best Yet!	4556	2021-01-19 20:12	140012	362

User_likes

user_id	video_id	reaction_type	reacted_at
2141	1529	LIKE	2012-04-12 04:46
2234	1529	LIKE	2012-04-30 17:47

Channel_user

channel_id	user_id	subscribed_datetime
376	2521	2018-02-06 20:43
376	2798	2018-07-10 06:12

GENRE

genre_id	genre_type
201	ACTION
202	COMEDY

Video_Genre

video_id	genre_id
1573	205
1574	205

1. Get all the videos with more than 1 lakh views.

Note:

Output must be in the alphabetical order of video name

Expected Output Format

video_id	name	duration_in_secs	published_datetime	no_of_views	cha
...	...	...	...	...	

SELECT video_id, name, duration_in_secs, published_datetime, no_of_views, channel_id FROM video WHERE no_of_views > 100000 ORDER BY name ASC;

2. Get videos from TEDx channel (id=353) with more than 50 thousand views.

Note:

Sort the output in the descending order of no_of_views and in the ascending order of video name

Expected Output Format

video_id	name	duration_in_secs	no_of_views
...	...	...	...

SELECT video_id, name, duration_in_secs, no_of_views FROM video WHERE channel_id = 353 AND no_of_views > 50000 ORDER BY no_of_views DESC, name ASC

3. Get the top 10 most viewed videos till date.

Note:

Sort the output in the descending order of no_of_views and published_datetime

Expected Output Format

name	channel_id	no_of_views
...	...	...

SELECT name, channel_id, no_of_views FROM video ORDER BY no_of_views DESC, published_datetime DESC LIMIT 10

4. Get all the recent movie trailers that have more than 1 lakh views.

Note:

- Consider the videos that have "trailer" in their `name` as trailers.
- Sort the output in the descending order of `no_of_views` and `published_datetime`

Expected Output Format

name	channel_id	no_of_views	published_datetime
...	...	...	...

SELECT name, channel_id, no_of_views, published_datetime FROM video WHERE no_of_views > 100000 AND name LIKE '%trailer%' ORDER BY no_of_views DESC, published_datetime DESC

5. Get all the videos that are released in the year 2018.

Note:

Sort the output in the descending order of

`published_datetime` and then in the alphabetical order of `name`

Expected Output Format

video_id	name	duration_in_secs	no_of_views
...	...	...	...

SELECT video_id, name, duration_in_secs, no_of_views FROM video WHERE strftime("%Y", published_datetime) = '2018' ORDER BY published_datetime DESC, name ASC

6. Get the distinct ids of videos that belong to the following genres.

genre_id	genre
201	Comedy
202	Action
203	Thriller
211	Scifi

Note:

Sort the output in the descending order of

video_id

Expected Output Format

video_id

...

SELECT DISTINCT(video_id) FROM video_genre WHERE genre_id IN (201, 202, 203, 211) ORDER BY video_id DESC

7. Get all the esport videos that crossed one lakh views and were released between 2018 and 2020.

Note:

- Consider the videos that have "esport" in their name as gaming videos.
- Sort the output in the descending order of no_of_views and published_datetime

Expected Output Format

name	published_datetime	no_of_views
...	...	...

SELECT name, published_datetime, no_of_views FROM video WHERE (cast(strftime('%Y', published_datetime) AS integer) BETWEEN 2018 AND 2020 AND no_of_views > 100000) AND name LIKE '%esport%' ORDER BY no_of_views DESC, published_datetime DESC

8. Get the total number of channels in the database as

channels_count

Expected Output Format

channels_count
...

SELECT count(channel_id) AS channels_count FROM channel

9. Get the highest and least number of views for the videos in the database.

Expected Output Format

highest_number_of_views	least_number_of_views
...	...

highest_number_of_views least_number_of_views

SELECT max(no_of_views) AS highest_number_of_views, min(no_of_views) AS least_number_of_views FROM video

10. Get the average number of views for the videos released by the "Single Shot" Channel (id = 373)

Expected Output Format

avg_views

...

SELECT avg(no_of_views) AS avg_views FROM video WHERE channel_id = 373

11. Get the total number of premium users in the platform as

premium_users_count .

Expected Output Format

premium_users_count

...

SELECT count(name) AS premium_users_count FROM user WHERE premium_membership = 1

12. Get the number of male and female premium users in the platform.

Expected Output Format

gender	total_users
F	...
M	...

SELECT gender, sum(premium_membership) AS total_users FROM user GROUP BY gender