Description | Discussion

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_o
...	...	...	

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

genre_id	genre

QUERY.SQL(Q5)

```sql
SELECT
  video_id,
  name,
  duration_in_secs,
  no_of_views
FROM
  video
WHERE
  cast(strftime("%Y", published_datetime) AS int) = 2018
ORDER BY
  published_datetime DESC,
  name ASC;
```

SHOW TABLES

RESET CODE RESET DB RUN CODE SUBMIT

QUERY RESULT

Description Discussion

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

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	publish
...	...	...	

QUERY.SQL(Q4)

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

SHOW TABLES

RESET CODE RESET DB RUN CODE SUBMIT

QUERY RESULT

Description | Discussion

...	...	...	

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

Note:

Sort the videos by `no_of_views` from highest to lowest. For videos with the same number of views, order them by `published_datetime`, with the most recent video first.

Expected Output Format

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

QUERY.SQL(Q3)

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

SHOW TABLES

Description | Discussion

Expected Output Format

video_id	name	duration_in_secs	publish
...	...	...	

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_o
...	...	...	

QUERY.SQL(Q2)

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

SHOW TABLES

RESET CODE | RESET DB | RUN CODE | SUBMIT

QUERY RESULT

Description | Discussion

...	...	...

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_o
...	...	...	

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

Note:

QUERY.SQL(Q2)

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

SHOW TABLES

RESET CODE | RESET DB | RUN CODE | SUBMIT

QUERY RESULT

QUESTIONS

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	publish
...	...	...	

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

Note:

QUERY.SQL(Q1)

```
SELECT
  *
FROM
  video
WHERE
  no_of_views > 100000
ORDER BY
  name ASC;
```

SHOW TABLES

RESET CODE RESET DB RUN CODE SUBMIT

QUERY RESULT

Description Discussion

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
...

QUERY.SQL(Q6)

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

SHOW TABLES

RESET CODE RESET DB RUN CODE SUBMIT

QUERY RESULT

...

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
...	...	...

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

QUERY.SQL(Q7)

```sql
SELECT
  name,
  published_datetime,
  no_of_views
FROM
  video
WHERE
  no_of_views > 100000
  AND name LIKE "%esport%"
  AND CAST(strftime("%Y", published_datetime) AS INTEGER) BETWEEN 2018
  AND 2020
ORDER BY
  no_of_views DESC,
  published_datetime DESC;
```

SHOW TABLES

RESET CODE RESET DB RUN CODE SUBMIT

QUERY RESULT

Description Discussion

Expected Output Format

premium_users_count
...

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

Expected Output Format

gender	total_users
F	...
M	...

```sql
SELECT
  gender,
  COUNT(*) AS total_users
FROM
  user
WHERE
  premium_membership = 1
GROUP BY
  gender;
```

QUERY.SQL(Q12)

SHOW TABLES

RESET CODE RESET DB RUN CODE SUBMIT

QUERY RESULT

Submit Feedback PROCEED TO NEXT

Description | Discussion

Expected Output Format

avg_views
...

11. Get the total count of premium users on the platform as `premium_users_count` .

Note - Consider those users as premium which has 1 in `premium_membership` column.

Expected Output Format

premium_users_count
...

QUERY.SQL(Q11)

```sql
SELECT
  count(premium_membership) AS premium_users_count
FROM
  user
WHERE
  premium_membership = 1;
```

SHOW TABLES

RESET CODE | RESET DB | RUN CODE | SUBMIT

QUERY RESULT

Description Discussion

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
...	...

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

Expected Output Format

avg_views
...

11. Get the total count of premium users on the platform as

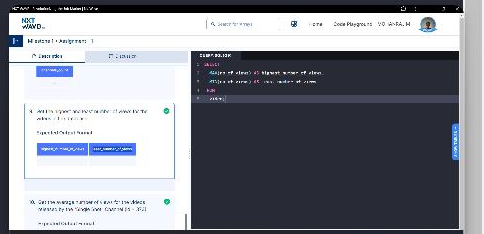
QUERY.SQL(Q10)

```sql
SELECT
  AVG(no_of_views) AS avg_views
FROM
  video
WHERE
  channel_id = 373;
```

SHOW TABLES

RESET CODE

QUERY RESULT



Snipping Tool

Screenshot copied to clipboard and saved
Select here to mark up and share.

Description Discussion

channels_count
...

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
...	...

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

Expected Output Format

avg_views
...

QUERY.SQL(Q9)

```sql
SELECT
  MAX(no_of_views) AS highest_number_of_views,
  MIN(no_of_views) AS least_number_of_views
FROM
  video;
```

SHOW TABLES

RESET CODE RESET DB RUN CODE SUBMIT

QUERY RESULT

Description | Discussion

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

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

Expected Output Format

channels_count
...

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

Expected Output Format

QUERY.SQL(Q8)

```sql
SELECT
  COUNT(*) AS channels_count
FROM
  channel;
```

SHOW TABLES