

INFO20003 Semester 2, 2023

Assignment 2: SQL

Due: Week 8 - Saturday 16th September 2023, 5:59pm

Submission: Via LMS <https://canvas.lms.unimelb.edu.au/>

Case: “MewTube” App

Description

As fellow Database experts, cat lovers, and social media enthusiasts, you and your classmates have created a start-up like YouTube called MewTube (inspired by the Pokemon character!). MewTube is a modern social video platform which hosts videos and enables content creators to have a friendly, less polarising, environment to engage with their fans; and allows content creators to collaborate on new videos.

For each user, MewTube records their details such as username, one email address, a login mechanism (which is defined strictly as one of the following: Google, Apple, Facebook, GitHub), and a reputation score (which is an integer from 0-100 inclusive, 100 as highly trustworthy and 0 being highly untrustworthy).

Each user can be optionally linked with a content creator account if they produce videos. Each content creator has an id, real name, screen name, and optional website. In addition, a content creator can have hashtags to describe themselves such as #music, #news, #memes, etc. For each content creator, MewTube tracks their videos. Content creators who collaborate on a video (e.g., MrBeast x BTS) are known as co-creators.

For each video, MewTube stores its id, title, upload timestamp, two long URLs storing the video object (the actual video data) and thumbnail image respectively, and a view counter. As above, each video is linked to its co-creators (which might just be a single content creator for solo-authored content). Same as each content creator, each video can be associated with a few hashtags, again, e.g., #news, #BTS, #experiment, #viral, etc. Also, each video might have some annotations – which are links that appear in a video that links to other videos.

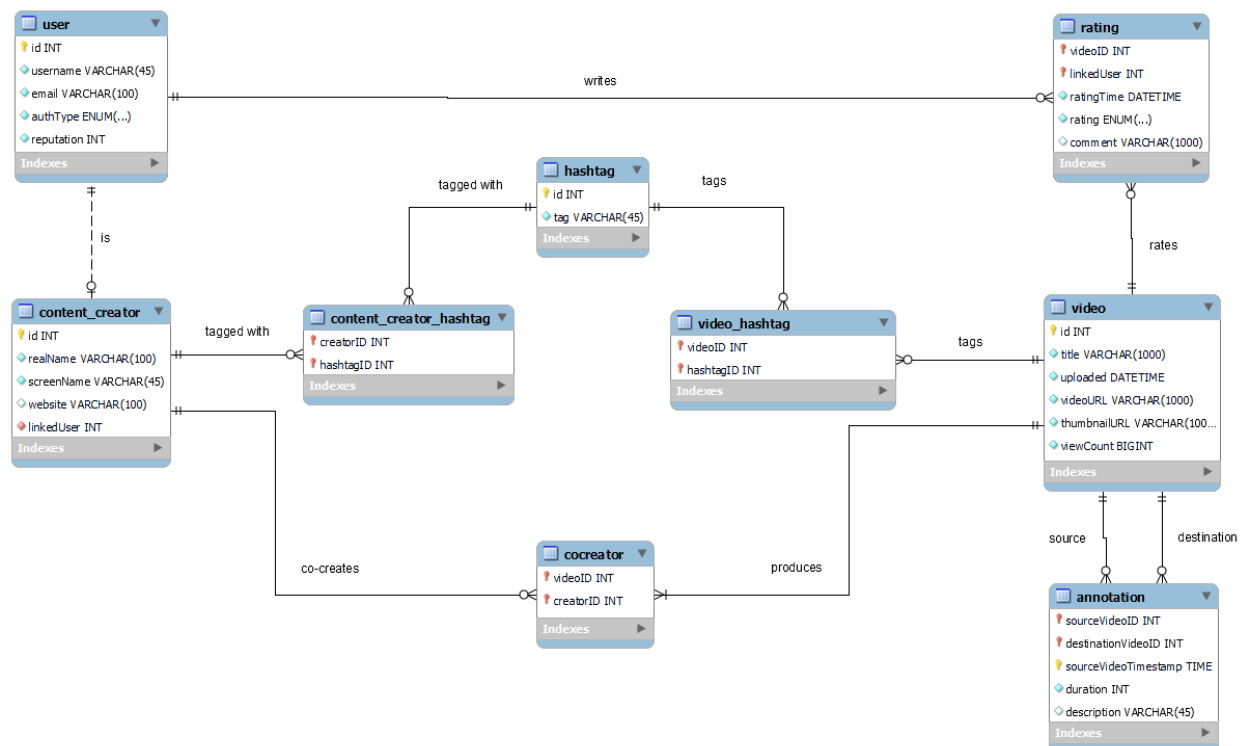
Each annotation identifies its source video (i.e., where it is seen), a destination video (i.e., the connection to another video), timestamp (when it pops up on the source video), duration (how long it appears on screen, in seconds), and description (text that is in the pop up).

Finally, each video has ratings that are left by other users. Each ratings record consists of a rating – i.e., either one of {Dislike, Neutral, Like}; a timestamp; and optional comment for the rating.

The Data Model

The Data Model from MySQL Workbench is provided in Figure 1.

FIGURE 1. DATA MODEL FOR MEWTUBE.



Assignment 2 Setup

Please pay special attention to the penalties listed [⚠].

A dataset is provided which you can use when developing your solutions. To set up the dataset, download the file **mewtube.sql** from the Assignment link on Canvas and run it in Workbench. This script creates the database tables and populates them with data. Note that this dataset is provided for you to experiment with: but it is not the same dataset as what your queries will be tested against (the schema will stay the same, but the data itself may be different). **This means when designing your queries, you must consider edge cases even if they are not represented in this particular data set.**

The script is designed to run against your account on the Engineering IT server (info20003db.eng.unimelb.edu.au). If you want to install the schema on **your own MySQL Server installation, uncomment the lines at the beginning of the script.**

⚠ WARNING: Do NOT disable only_full_group_by mode when completing this assignment. This mode is the default and is turned on in all default installs of MySQL workbench. You can check whether it is turned on using the command `SELECT @@sql_mode;`

The command should return a string containing `ONLY_FULL_GROUP_BY` or `ANSI`.

When testing, our test server WILL have this mode turned on, and if your query fails due to this, you will lose marks.

The SQL Tasks

Please pay special attention to the penalties listed [⚠].

In this section are listed 10 questions for you to answer. Write one (single) SQL statement per question. Each statement must end with a semicolon (;). Subqueries and nesting are allowed within a **single** SQL statement – however, you may be penalised for writing overly complicated SQL statements.

⚠ WARNING: DO NOT USE VIEWS (or 'WITH' statements/common table expressions) OR VARIABLES to answer questions. Penalties apply.


Some Clarifications

- Hashtags can be applied to both content creators and videos. A video's tags do not necessarily match the tags of the content creator(s) who created the video. When hashtags are involved, the question will inform you if you need to look at content creator tags or video tags.
- In the hashtag table, the hash symbol (#) is included in the tag string. For example, the hashtag #food is stored as the string '#food' (not just 'food').






? The Questions

1. List all videos which contain no annotations (i.e., find videos that do not have an annotation linking to another video). Your query should return results of the form (videoID, title). **(1 mark)**
2. Find the most recent user rating record in the entire database. Assume there are no ties (only one is the most recent). Your query should return results of the form (videoID, username, ratingTimestamp). **(1 mark)**
3. List all videos created by content creator TaylorSwiftOfficial that have at least 1 million views. Note that 'TaylorSwiftOfficial' is the screen name of the account. Your query should return results of the form (videoID, title). **(1 mark)**
4. Find the video which is most linked to (i.e., appears the most as 'destination video' for annotations). If there are ties, then you must return all videos with the highest number. Your query should return results of the form (videoID, title, linkedCount), with one row per video in case of a tie. **(2 marks)** `MAX(COUNT(`
5. List the upload datetime for the videos that have #memes as a hashtag and have been rated at least 3 times. Your query should return results of the form (videoID, uploadDatetime, ratingCount). **(2 marks)**
6. Find the names of controversial content creators, defined as users who have < 50 reputation, but have at least 3 videos, and at least 6 ratings given to their videos in total. Your query should return results of the form (username, realName, screenName). **(2 marks)**
7. Find which hashtag has the highest number of polite comments made to videos using that hashtag. Polite comments are comments that contain 'thank you' or 'well done' (you can ignore the casing of these phrases). Ignore hashtags in content creator profiles. If there are ties, then you must return all results. Your query should return results of the form (hashtag, commentCount), with one row per hashtag in case of a tie. **(2 marks)**
8. List the top 3 hashtags with the highest total annotations as a destination video. Also return their total duration in annotations. Your query should return results of the form (hashtag, totalAnnotationsAsDestination, totalDuration). If there are ties in the top 3 positions, you must return all ties. For example, let's say the database contains seven hashtags and the annotation counts for each hashtag are (5, 4, 4, 3, 3, 2, 1). The top 3 counts are 5, 4 and 3 so you need to return the top 5 rows, which are the ones having annotation counts of (5, 4, 4, 3, 3). **(3 marks)**
9. Find the content creators whose own hashtags include '#memes' who have co-created at least one video with at least one other creator whose hashtags contain '#technology'. (Note: do NOT consider the hashtags of the videos themselves). Note that we only want to consider co-created videos where the #memes creator is distinct from the #technology creator. To elaborate: if MrBeast is a #memes and #technology creator, we need a collaboration with a different creator who has a #technology hashtag for MrBeast to be included in the results. Your query should return results of the form (realName, screenName) of the content creators associated with #memes. **(3 marks)**
10. Find the content creators who have not co-created a video before the start of this year (01/01/2023) with the creator INFO20003Memes but have co-created at least one video with INFO20003Memes on or after 01/01/2023 (i.e., new co-creator partnerships on or after 01/01/2023). Note that 'INFO20003Memes' is the screen name of the account. Your query should return results of the form (realName, screenName) for all such creators. Do not return a row for INFO20003Memes. **(3 marks)**


SQL Response Formatting Requirements

Please pay special attention to the penalties listed [].

To help us mark your assignment queries as quickly/accurately as possible, please ensure that:

- Your query returns the projected attributes in the same order as given in the question and does **not** include additional columns.
 - E.g., if the question asks, 'return as (userId, name)', please write `SELECT userId, name ...`
 -  **DO NOT return attributes in the WRONG order**, e.g., `SELECT name, userId...`
 - You can name the columns using ``AS`` however you'd like, only the **order** matters. E.g., this is fine: `SELECT userId, name AS fullName`
- Please do NOT use "databaseName.tableName" format.
 - E.g., please write `"SELECT userId FROM users..."`
 -  **DO NOT provide the database name**, e.g. `SELECT userId FROM coltonc.users ...`
- Ensure that you are using single quotes (`'`) for strings (e.g. `...WHERE name = 'bob'...`) and double quotes (`"`) only for table names (e.g. `SELECT name FROM "some table name with spaces"...`) .
 -  **Do NOT use double quotes for strings**: `...WHERE name = "bob"...`
 -  **Do NOT use Microsoft Word 'smart quotes'** (the fancy ones as you see in "this" 'example').
- Comments are optional, but we recommend writing them for complex queries.
-  **Do NOT delete the special comment markers in the SQL template file**. These include: `-- BEGIN QX`, `-- END QX`, and `-- END OF ASSIGNMENT` (where X is the question number). They help us mark your submission so tampering with them will hinder our marking and may attract penalties.


Assignment Submission Instructions

Please pay special attention to the penalties listed [].

Your submission will be in the form of an SQL script. There is a template file on the LMS, into which you will paste your solutions and fill in your student details (more information below).

This .sql file should be submitted on Canvas by **the time indicated on the first page of these instructions**.

Name your submission as 987654.sql, where 987654 corresponds to YOUR student ID number.

Please make sure that you **actually submit** your file on Canvas . After uploading the file, you need to press **'Submit Assignment'** to actually submit the file. If you submit late because you failed to press the submit button and only noticed this after the deadline, your submission will be considered late just like any other late submission to maintain fairness for all students.

Filling in the template file:

The template file on the LMS has spaces for you to fill in your student details and your answers to the questions. There is also an example prefilled script available on Canvas as well.

Below (Table 1) are screenshots from those two documents explaining the steps you need to take to submit your solutions:

TABLE 1: SCREENSHOT EXAMPLES ON HOW TO SUBMIT THE SOLUTIONS.

Step	Example
1. At the top of the template, you'll need to replace "XXXXXXX" with your student number and name	<p>Template</p> <pre>-- Your Name: XXXXXXXX -- Your Student Number: XXXXXXXX</pre>
	<p>Example Filled in</p> <pre>-- Your Name: Colton Carner -- Your Student Number: 693281</pre>
2. For each question 1-10, place your SQL solution in between the "BEGIN QX" and "END QX" markers. <u>Ensure each query is terminated with a semicolon ";"</u>	<p>Template</p> <pre>-- -- BEGIN Q1 -- -- END Q1 --</pre>
	<p>Example Filled in</p> <pre>-- -- BEGIN Q1 --**This is an example of a comment which will not be executed --**(comments are not NEEDED, but if your query is complex you can leave some) --**Below is an example of how you would enter your answer: SELECT * FROM delivery NATURAL JOIN deliveryitem WHERE supplierid = 101; --**It's OK to add more space in between the 'BEGIN QX' and 'END QX' markers --**for each question, just don't DELETE the markers! --**Make sure you fill out your name + student num at the top of the document --**After reading / understanding these comments in the Q1 section --**(comments starting with '**'), feel free to delete them --**(don't delete lines without **) -- END Q1 --</pre>

3. Test that your script is valid SQL by running it from MySQL Workbench. Run the entire script by copy-pasting this entire file into a new workbench tab, placing your cursor at the start of the file (without selecting anything), and pressing the lightning bolt to run the entire file.



All queries should run successfully one after another. If not, check to make sure you added semicolons ';' after each query.

All 10 queries ran sequentially and were successful.

	#	Time	Action
✓	9	13:15:53	select id, t...
✓	10	13:15:53	select foru...
✓	11	13:15:53	select foll...
✓	12	13:15:53	select ad...
✓	13	13:15:53	select out...
✓	14	13:15:53	select pos...
✓	15	13:15:53	select par...
✓	16	13:15:53	select id fr...
✓	17	13:15:53	select out...
✓	18	13:15:53	SELECT ...

Late submission

Unless you have an approved extension (see below), **you will be penalised -10% of the maximum number of marks in the assignment per calendar day that your submission is late** ⚠. For instance, if you received a 78% raw score, but submitted 2 days late, you'd receive a 58% for the assignment.

Requesting a Submission Deadline Extension

If you need an extension due to a valid reason, you will need to provide evidence to support your request by 6 pm on Friday 15 September 2023. Requests received after this time may be rejected. Medical certificates need to be at least two days in length.

To request an extension:

- Email Timothy Hermanto (timothy.hermanto@unimelb.edu.au) from your university email address, supplying your student ID, how many days you'd like to extend, and evidence that can support the number of days you are requesting. Please include in the subject [INFO20003 Assignment 2 Extension].
- If your submission deadline extension is granted you will receive an email reply granting the new submission date. **Do not lose this email!**

Reminder: INFO20003 Hurdle Requirements

To pass INFO20003, you must pass two hurdles:

- **Hurdle 1:** Obtain at least 50% (15/30) or higher for the three assignments (each worth 10%)
 - **Hurdle 2:** Obtain at least 50% (35/70) or higher for the combination of quizzes and end of semester exam
- It is our recommendation to students that you attempt every assignment and every question in the exam.

GOOD LUCK!