# INFO20003 Semester 1, 2022

**Assignment 2:** SQL
**Due:** 6:00pm  Friday 29th April
**Weighting:** 10% of your total assessment

## newQuora Startup Database

## Description

You and a group of fellow undergrads have created a start-up called 'newQuora'. The company's goal is to create an online question-and-answer forum for users around the world.

The system has two kinds of users: general and admins. Most user attributes are straightforward personal information and are listed in the ER diagram below.
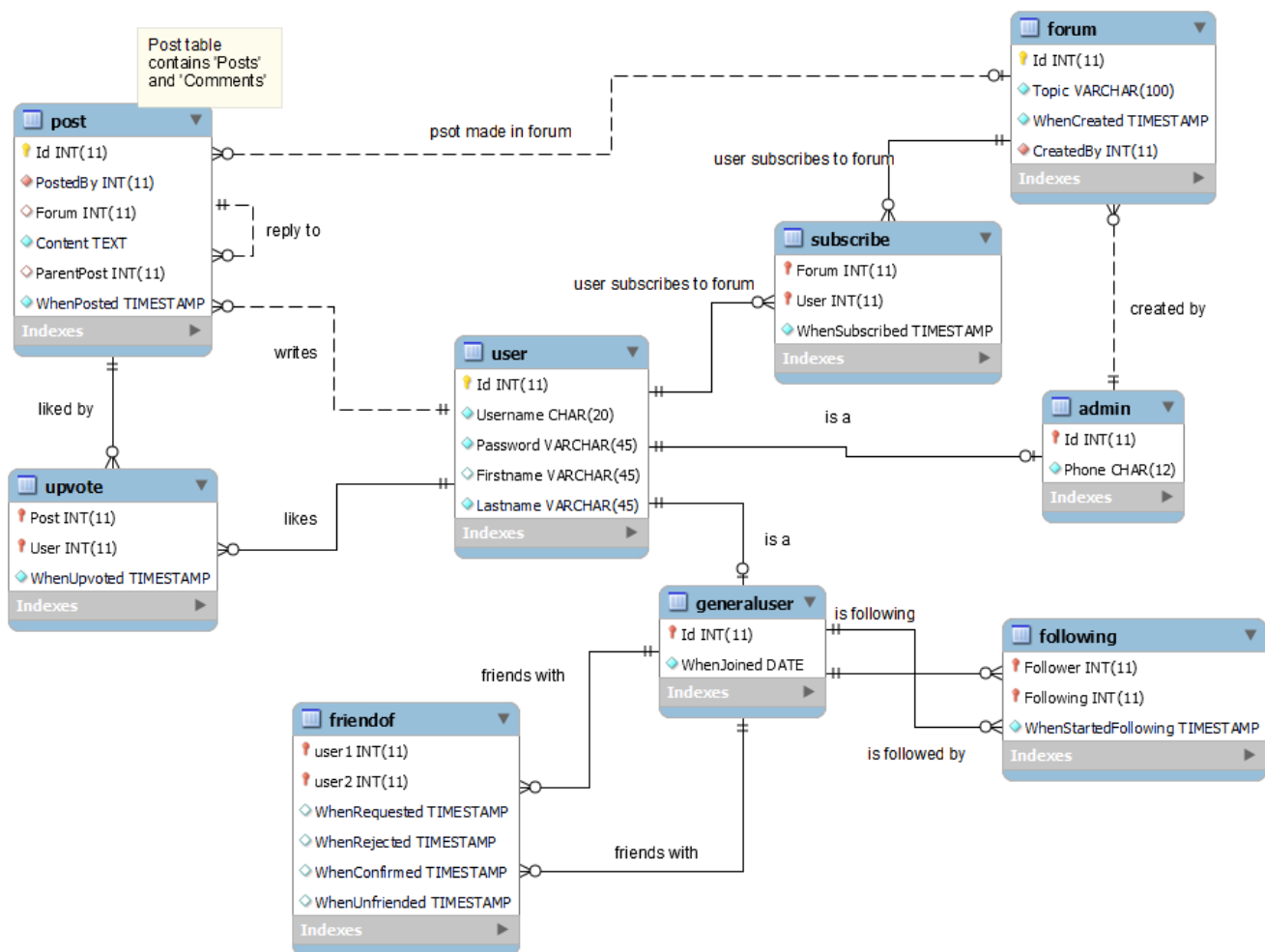
Discussions are organized into forums, each of which is about a particular topic. New forums can be opened by admins. Users can subscribe to any number of forums to get regular updates.

Users can create a 'post' in any forum, which becomes a topic for discussion. Any user can comment on a post, or on comments of a post (nested comments). Users can also "upvote" a post or comment.

General users (but not admins) can have different "relationships" among themselves. One user can be "following" another to receive updates when they post or comment. *Note that following is a non-symmetric relation: if A is following B, that does not imply B is following A. A is denoted as a "follower" of B.*

General users (but not admins) can also add each other to friend-lists. When one user sends a friend-request to another, the latter can reject or accept the friendship. If the latter accepts, the pair are now friends. Note that friendship is a symmetric relation: if A is a friend of B, then B is a friend of A, whether A sent the friend request to B or vice-versa. Once a pair of users are friends, either may later unfriend the other, in which case the friendship ends for both. A user's "friends" means their current "confirmed" friends, not those where the friendship has ended or not begun.
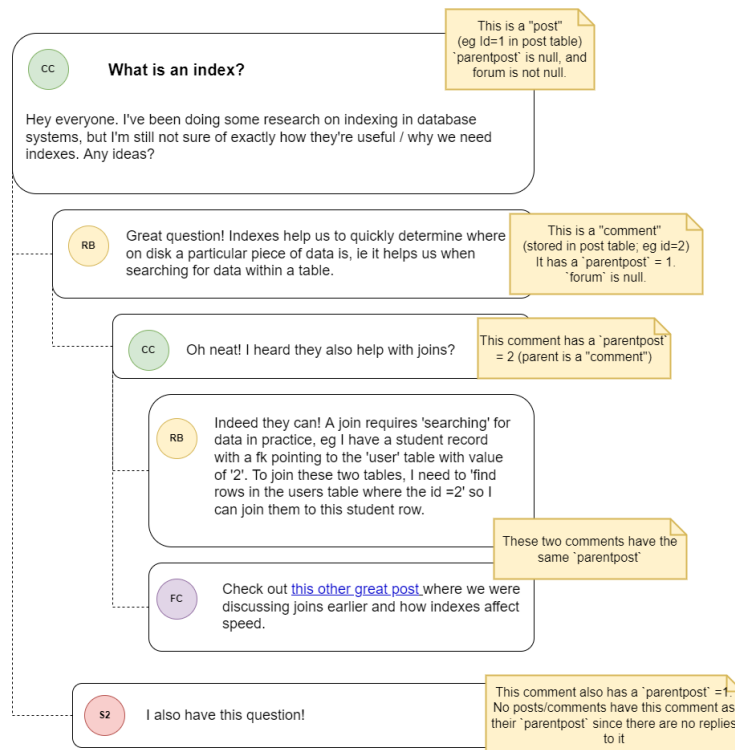
# The Data Model



*The physical ER model of newQuora startup database.*

# Notes on Implementation

**Posts + Comments:**

- Comments are stored in the same table as Posts, and are connected to their parent posts/comments via a unary relationship.
- Posts have a non-NULL FK to the forum they are posted in, and a NULL FK for their 'parentpost'.
- Comments have NULL for forum FK (since they're not a "forum" post), and a non-null 'parentpost' FK.
- Both posts and comments can be 'parentposts', i.e. users can comment on posts or on another comment. A post/comment can be the 'parentpost' of many comments, but a comment will only have one 'parentpost'.

*An example of a forum with post + comments, and how this relates to the `parentpost` and `forum` attributes in the post table.*

**Friendships**:

- We record when a friend request is made, and accepted/rejected/confirmed in the FriendOf table. newQuora does not allow you to send a friend request to a user that you have rejected or unfriended, and there is no way for the other person to send another friend request to you if you have rejected or unfriended them.
- `User1` in the friendOf table was the user that sent the friend request.

# Assignment 2 Setup

A dataset is provided against which you can use when developing your solutions. To set up the dataset, download the file **newQuora_2022.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.*

***Note: 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

In this section are listed 10 questions for you to answer. Write one (single) SQL statement per question. Subqueries and nesting are allowed within a single SQL statement – however, you may be penalized for writing *overly* complicated SQL statements. DO NOT USE VIEWS (or 'WITH' statements/common table expressions) to answer questions.

1. List all users who have sent a friend-request which is still 'pending' (i.e. a friend request has been sent but not confirmed or rejected yet). Your query should return results of the form (userID) **(1 mark)**

2. List all the forums that have at least one subscription. Your query should return results of the form (forumId, topic, numSubs). **(1 mark)**

3. Find the forum with the most recent post. You may assume there are no ties, i.e. there is only one forum with the most recent post. Your query should return one row of the form (forumId, postId, whenPosted)." **(1 mark)**

4. For each user with at least one follower, list the IDs of all their followers. Your query should return results of the form (userId of followed, userId of follower). E.g. If user #1 is followed by users #2 and #3, you'd return two rows: (1,2) and (1,3). **(2 marks)**

5. Which forum has the highest number of upvotes on its posts (sum of upvotes of all posts in that forum), and which admin created that forum? Assume there are no ties for first place, and that at least one forum has a post with at least one upvote. Return one row as (adminId, forumId, numberOfUpvotesInForum). **(2 marks)**

6. List all the users who have no "followers" (i.e., no other user is "following" them) and no friends. Your query should return results of the form (userId, username).    **(2 marks)**

7. A "high-quality contributor" is a user who has a mean average of at least one upvote for every post or comment they've made, and has made at least one post or comment. Find all of the high-quality contributors and their average upvotes. Return as (userId, avgUpvotes). **(2 marks)**

8. Find all comments or posts that have fewer likes than every comment that they are the parentpost of. (I.e. every direct reply/comment to the parent post/comment has more upvotes than the parent). Return as (PostOrCommentId). **(3 marks)**

9. List all the 'general' users who have only upvoted posts or comments made by an admin OR someone who is currently their friend (i.e. they don't like any posts of someone who is not an admin and not currently their friend). Users who have liked their own posts should not be returned. Your query should return results of the form (userID).  **(3 marks)**

10. For each admin, show the forum with the highest number of subscriptions that they have created. If an admin has created multiple forums with an equal highest number of subscriptions, you must return all forums with this tied highest value. Your query should return results of the form (adminID, forumId, numSubscriptions), one row per admin (unless there are ties). *If an admin has never created a forum, return (adminId, NULL, 0).* **(3 marks).**

# SQL Response Formatting Requirements

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 …"** instead of **"SELECT name, userId…"** (*you can name the columns using* `AS` *however you'd like, only the **order** matters*).

- Please do NOT use "databaseName.tableName" format. E.g., please write **"SELECT userId FROM users…"** instead of **"SELECT userId FROM coltonc.users …"**.

- Ensure that you are using single quotes( ' ) for strings (e.g. **…WHERE name = 'bob'…**) and double quotes ( " ) <u>only</u> for table names (e.g. **SELECT name FROM "some table name with spaces"…**) . Do **NOT** use double quotes for strings **"…WHERE name = "bob"…"**.
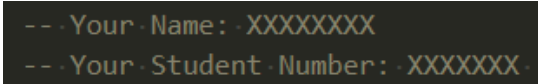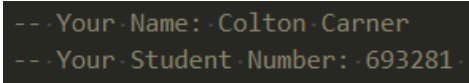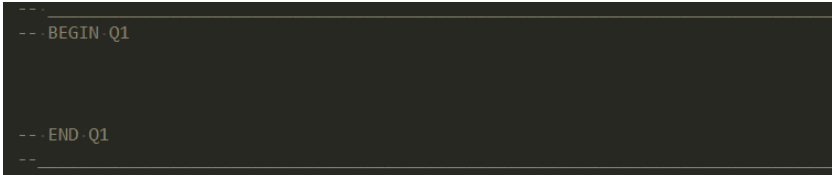
# Submission Instructions

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 **6pm** on the due date of **Friday, April 29th**. Name your submission as 987654.sql, where 987654 corresponds to YOUR student id.

### 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 the LMS as well. Below are screenshots from those two documents explaining the steps you need to take to submit your solutions:

| Step | Example |
|---|---|
| 1. At the top of the template, you'll need to replace "XXXXXXX" with your student number and name | *Template*<br><br>`-- Your Name: XXXXXXX`<br>`-- Your Student Number: XXXXXXX` |
| | *Example Filled in*<br><br>`-- Your Name: Colton Carner`<br>`-- Your Student Number: 693281` |
| 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> | *Template*<br><br>`--`<br>`-- BEGIN Q1`<br><br><br>`-- END Q1`<br>`--` |
| | *Example Filled in* |

```
-- .
-- .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
-- .
```

| | 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 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 total number of marks in the assignment per 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% score for the assignment.

## Requesting a Submission Deadline Extension

If you need an extension due to a valid (medical) reason, you will need to provide evidence to support your request by **5pm on the 28th of April**. Extensions received after this time may be rejected. Medical certificates need to be at least two days in length.

To request an extension:

- Email Farhana (farhana.choudhury@unimelb.edu.au) from your university email address, supplying your student ID, the extension request and supporting evidence. Please add INFO20003 in the subject title.
- 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) for the three assignments (each worth 10%)
- **Hurdle 2:** Obtain at least 50% (35/70) for the combination of the quizzes and final exam

Therefore, it is our recommendation that you attempt every assignment and question in the exam.

# GOOD LUCK!