# Assignment 2: SQL

INDIVIDUAL PROJECT (10%)

Case

You are the SQL programmer for a mental wellbeing app, which allows users to do two types of things. The first is to participate in a Twitter-style social network. The second is to do online units called Steps, which are interactive therapy modules designed to exercise and develop a range of psychological skills. Your job is to write SQL queries that allow the app to work. The data model is as follows:
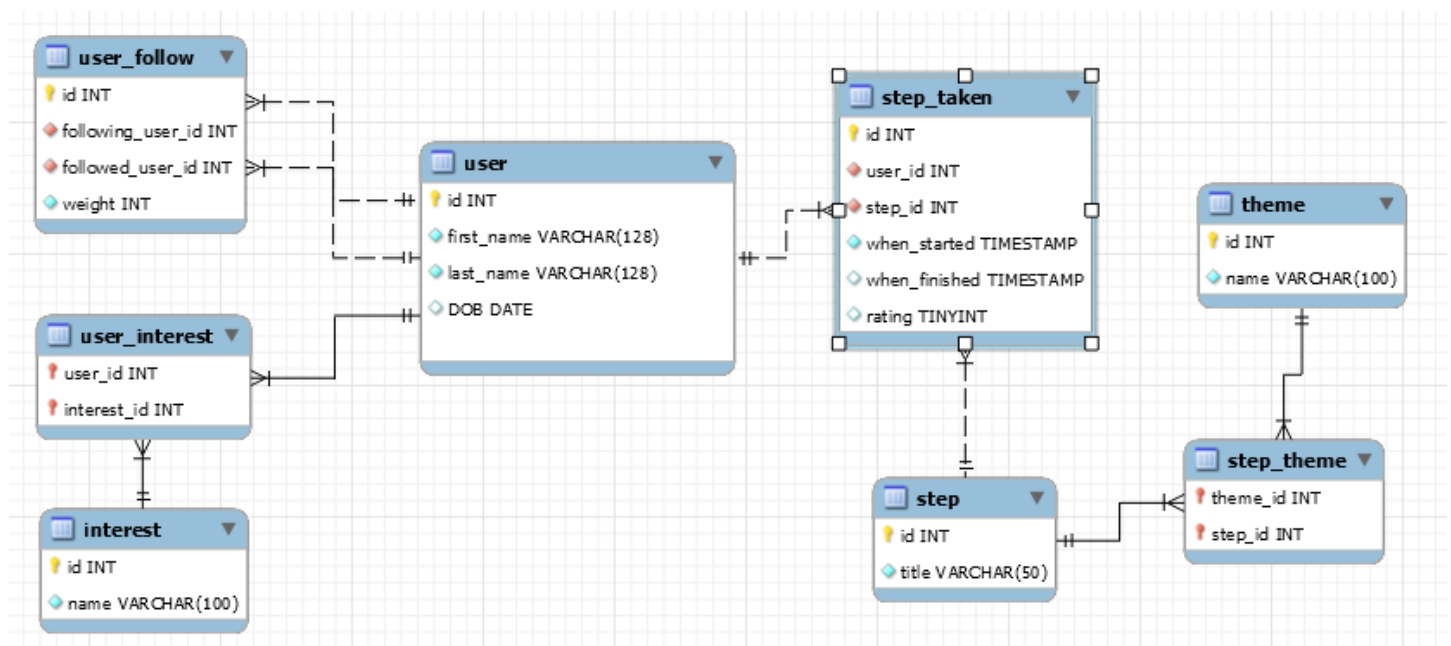
How the app works

When users register into the system, a record is created in the `user` table. Upon registering, a user can follow other users to see their posts and activity. If user *x* (following) starts following user *y* (followed), an appropriate entry is created in the `user_follow` table (*x* following *y* does not necessarily mean that *y* follows *x*). The weight attribute of this table holds an integer that reflects the extent to which *x* follows *y*; for example, this number would increase the more *x* likes or comments on posts created by *y*. To help connect users, each user can also select, from a list stored in the `interest` table, interests that they have (e.g. movies, music, books). For each interest that a user selects, a record is created in the associative table `user_interest`.

On the therapy side of this app, users can do online modules called Steps. Different Steps take different amounts of time to complete and users can do a Step in one sitting or return to a Step and complete it later on. Sometimes they might not finish a Step and will instead move on to something else. Also, they can do any given Step as many times as they would like and each time they finish a Step they can rate how it made them feel (1 being the lowest and 5 being the highest).

As soon as user *x* makes a start on doing Step *y*, an entry is created in the `step_taken` table to record the fact that *x* started doing *y*. The timestamp of this start is also automatically recorded in the `when_started` column. If a user finishes this attempt to do the Step, the datetime of completion is recorded in the `when_finished` column.

Steps are also categorised under themes, which are used to indicate what types of things a Step might help with. For example, a Step that helps with mindfulness might be categorised under the Mindfulness theme. The relationship between Steps and Themes is many-to-many.

Setup Script

To set up the database in your MySQL server, download the file asst2-2019s1-setup.sql from LMS and run it in Workbench. This script creates the database tables and populates them with test data.

Note the comments near the start of the script. If it is your own server, you will want to uncomment the uncomment section near the top of the script, so that you create a new schema ('wap') to store the tables in. You cannot create a new schema on the UniMelb Engineering IT server (info20003.unimelb.edu.au).

The SQL queries you must write

Over the page are listed 10 questions for you to answer. Write *one* SQL statement to answer each question. Do *not* use *views* or *temporary tables* to answer the questions.

Beside each question is a maximum mark which reflects the difficulty of the question. Your total score will be scaled to 10% of your overall marks in the subject.

Make sure your code and output are readable – this will affect your mark. Where it would improve readability, order your output and use aliases. Format large numbers and fractions appropriately.

Assessment

Your work will be assessed according to three criteria:

- Correctness of output (70%)
- Simplicity of code (20%)
- Correctness of formatting (10%)

The "simplicity" requirement means that your code should not be unnecessarily complex. For example, a query that produces correct output, but say, joins more tables than is necessary, may not achieve full marks, even though it produces the right output.

INFO90002 hurdle requirements

To pass INFO90002 you must obtain a grade of 35/70 for the exam and more than 15/30 for the two assignments combined.

The Questions

1. List the full names (e.g. Alice Smith), as one column, of the users who have not taken any Steps yet. (1)

2. List all of the IDs and titles of Steps that contain the substring 'mindful' in their title. (1)

3. Provide a list of the titles of all Steps *completed* by user with id = 17. Do not show duplicates (list each title only once). (2)

4. Provide a list of the titles of all Steps that have been taken more than two times along with a count of how many times. (2)

5. Which Step(s), listed with columns id, title and the count of times taken, have been taken the greatest number of times? (3)

6. List each Step with the title column, along with a count of how many times that Step has been taken and the average rating received by the Step (formatted to 2 decimal places). Order the result by the average rating as a number in descending order. (3)

7. Provide a list of the titles of all Steps that have been taken by both Alice (id = 1) and Bob (id == 2), along with the combined number of times they have taken the Step. (4)

8. List users older than or equal to 21 years of age, along with a count of how many other users they are following and a count of how many other users are following them. List the user's id, first name, last name, age, following count and followed count, and order the results by first name ascending, then last name ascending. (4)

9. For each (user, theme) pair such that user has taken some steps under the theme, provide a count of how many times a user has taken a step that is categorised under the theme. The output should consist of user ID, user first name, user last name, theme name and the count of steps taken. (5)

For question 10, you can attempt one and only one of 10A or 10B:

10.

A) Provide a complete list of all user ID pairs such that the two users follow each other and share at least one interest. (Hint: MySQL has a CROSS JOIN operator, which returns the Cartesian product of rows from the joined tables) (5)

B) A query that returns a row for each instance in which two users, *x* and *y*, share an interest *z*. The result should consist of user *x* ID, user *x* first name, user *y* ID, user *y* first name and the interest name. The returned results should

be such that only one of (*x*, *y*) or (*y*, *x*) is returned; for example, if row [1, Alice, 2, Bob, Tennis] is in the result set,

then [2, Bob, 1, Alice, Tennis] should not be. (5)

Submission

Submit a single PDF showing your ten answers to LMS by 6 pm on the due date of Friday 3rd May 2019, at the end of week 8.
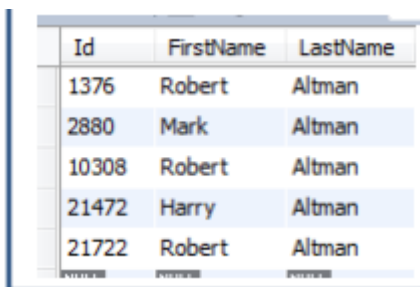
Ensure that you place your student number at the top of every page of your submission.

For each question, present an answer in the following format:

- Show the question number and question in black text.
- Show your answer (the SQL statement) in blue text (not a screen shot)
- Show a screenshot from Workbench showing output of 10 or fewer lines.
- Show how many rows were returned, in red text

Question 7. List all users with the last name 'Altman'

```
SELECT *
FROM user
WHERE lastname = 'Altman';
```

Id	FirstName	LastName
1376	Robert	Altman
2880	Mark	Altman
10308	Robert	Altman
21472	Harry	Altman
21722	Robert	Altman

5 Rows Returned

SQL queries must be formatted in an easy-to-read manner. This means writing keywords in all-caps,

placing most clauses on new lines, and indenting subqueries. For example, this *is acceptable*:

```
SELECT DISTINCT saleId
FROM Sale
WHERE departmentID IN (SELECT departmentId FROM Department WHERE floor = 2);
```

whereas this is not acceptable:

```
select distinct saleId FROM Sale where departmentID IN (SELECT departmentId fromcDEPARTMENT
WHERE Floor = 2);
```

Academic Honesty

This assignment must be your own work. Plagiarism - the copying of another's work without proper acknowledgment - is not permitted. Nor is allowing another person to copy your work. Work submitted for assessment purposes must be the independent work of the student concerned. Please refer to http://academichonesty.unimelb.edu.au/ for details about academic honesty.

Academic misconduct occurs when students portray someone else's work as their own. There are many ways in which academic misconduct can occur. Some of these are:

- Sham Paraphrasing: Material copied verbatim from text, with source acknowledged in-line,
- but represented as paraphrased.
- Illicit Paraphrasing: Material paraphrased from text without in-line acknowledgement of
- source.
- Other Plagiarism: Material copied from another student's assignment with the knowledge of
- the other student.
- Verbatim Copying: Material copied verbatim from text without in-line acknowledgement of
- the source.
- Recycling: Same assignment submitted more than once for different subjects.
- Ghost Writing: Assignment written by third party and represented as own work.
- Purloining: Assignment copied from another student's assignment or other person's papers without that person's knowledge.

The University is committed to graduating students with "a profound respect for truth, and for the ethics of scholarship... we want our graduates to be capable of independent thought, to be able to do their own work, and to know how to acknowledge the work of others" (Professor Peter McPhee).

As such, the university takes a dim view of students who are not able to correctly acknowledge the work of others, or who try to pass this work off as their own.

All students should check the web site http://academichonesty.unimelb.edu.au/ which provides practical advice to students about how to avoid academic misconduct.