Databases Formative Exercises, Spring 2019

These are formative exercises which means that they do not count towards your unit mark and you are not required to submit any solutions. The feedback that you will get on these exercises if you do submit should help you review your progress on this unit and obtain a better mark on coursework 1. These formative exercises cover material from weeks 13-14.

Exercise 1: modelling

For this exercise, you must submit a single PDF file containing a single A4 page. Anything else (e.g. word documents) will not be marked. To create an ER diagram, you can either draw the diagram by hand, scan it in and save as a PDF (uni printers can do this for you) or you can use diagramming software. draw.io is a free site that supports Crow's Foot notation (in the "Entity Relation" section).

Create an ER diagram in Crow's foot notation for the following scenario. All attributes are mandatory unless otherwise stated. Include the multiplicity of all relationships.

> The University of Bristol Hoverboard Society (HovSoc) wants to create a database to manage its membership and events. Each member has a name, an optional student number[1], a contact e-mail address and a hoverboard riding skill level (represented as an integer, minimum 0). We assume that e-mail addresses are unique among members.
>
> The committee consists of some of the members, each of which has a unique committee role. We assume that the database is reset every year, that committee roles do not change during the year and that each committee role must be filled every year.
>
> An event has a date, a name, a location, an optional description and an organiser who must be a society member (not necessarily a committee member). An event is attended by a set of members. There is never more than one event at the same location on the same date but event names are not unique[2].
>
> Members may have optional skills that are relevant to the society such as first aid qualifications, advanced hoverboard riding certificates or minibus driver training. Where a member has an optional skill, this is recorded at a particular level i.e. "David has first aid skills at level 4" (level is an integer, which means different things depending on the skill in question, always in the range 1–10). Skills are a separate entity as there are a number of "standard" ones that the society is interested in (i.e. first aid qualifications) even though some years they may not have any members with this skill.

[1] Some members are not students anymore. Don't mention this to the union.
[2] 2In fact, lots of events are named "Pub meet". These have a NULL description as it's obvious what they are and the name of the pub is in the location field.

Exercise 2: Create/Drop script

For this exercise you should submit a SQL script in a file with ending ".sql".

A create/drop script is an SQL file to create the tables in a database. Create/drop scripts are particularly useful during development and testing as you can quickly "reset" the database.

A create/drop script starts off by dropping all tables that are part of its schema with DROP TABLE IF EXISTS statements. This means that the script will run without errors[3] whether or not the tables already exist, so you can use the script both to set up the database from scratch and to "reset" it if something goes wrong.

After the DROP statements, a create/drop script creates all the tables for its schema with CREATE TABLE statements. The order of statements is important: if B has a foreign key to A then you have to create A before B and you have to drop B before A or you will get an error. Once you have found the correct order in which to create your tables, you can get the drop statements to work by putting them in the opposite order to the create statements.

If you want two tables A and B to both have foreign keys to each other, you will have problems whichever way round you try and write the CREATE statements. In this case you should redesign your schema or add a separate join table.

A create/drop script exists only to create and drop tables. It does not insert any data into the tables – if your tables should contain some initial data, this would go into a separate script.

The task of this exercise is simply to create a create/drop script for the "HovSoc" schema that you modelled in the last exercise.

Among other things this means that you need to choose where to place foreign keys or join tables to store the relationships in your schema.

In Spring 2019, you must submit your solutions to these exercises on Blackboard. Go to the unit page and select the "Formative Exercises" heading. Since you must upload two files (one PDF and one SQL script), in the window where you select the files to upload, you should select both files at once.

[3] MariaDB gives you a warning, which you can ignore.