

## Exercise 5 – Database Design and SQL

Weighting: 2%

---

The aim of this exercise is to introduce you to database design, SQL, and the use of the SQLite database management system.

### Task 1 – Produce Database Designs for the following problems:

- a) **Online blog example.** A blog management site has a set of blogs. Each blog has a title and a creator and a set of articles. Each article has a headline, a text and an author, and appears in a single blog.
- b) **Online newspaper example.** A newspaper has a set of (daily) issues. Each issue has a set of articles. Each article has a headline, a text, an author and a set of photos, and appears in a single issue. Each photo has a caption, a photographer and an image, and may appear in different articles.
- c) **Online journal example.** A journal has a name, a date, and a set of articles. Each article has a title, a date, a set of authors, a body, and a set of references (which are themselves articles). Furthermore, an article may be referenced by other articles. Each author has a name and an affiliation. An author may contribute to different articles.

We suggested tackle this problem in the following order:

1. Work out the tables and relationship between journal and articles.
2. Extend your solution to include authors (and the appropriate relationship authors have with other entities).
3. Extend your solution to include references.

When demonstrating your work to your reviewers, you need explain the relationships between entities (1 to many or many to many). You need to explain how these relationships are implemented in your designs.

### Task 2 – Write SQL queries for the following based on the blog example from Task 1:

- a) What are the titles of all blogs with "Japan" in the title?
- b) What are the titles of all blogs that contain an article with "Japan" in the headline?

c) What are the titles of all blogs that contain an article written by the blog creator?

When demonstrating your work to your reviewers, you need to explain how you join tables and how you handle partial matching.

### Task 3 – Creating a SQLite Database from the command line

Follow the lecture to create a SQLite database from the stockdatabase.sql file provided in the WebDev-Examples repository. Or you can follow the instructions below.

- Copy the **week5** directory to your directory containing your lab work.
- In your **week5** directory execute the following command:
  - **sqlite3 stockdb.sqlite**
- This launches sqlite3 and creates the stockdb database.
- Type the command **.help** to see the available commands.
- Import the SQL definition file **stockdatabase.sql** in the **week5** directory into your SQLite database:
  - **.read stockdatabase.sql**
- Note: this will only work if you launched sqlite3 in the same directory as the .sql file
- Check the tables you have imported with the following SQLite command:
  - **.tables**
- Execute some of the queries from the lecture notes (e.g. Page 21 of Lecture 5 slides) such as:
  - **SELECT \* FROM Stock;**
- Note: the Up and down arrows allow you to cycle through previous commands and edit them.
- To **export** all or part of an SQLite database, use the **.dump** command:
  - **.dump tablename**
- If tablename is omitted, all tables in the database are exported.

### Task 4 – Write the following queries for stockdatabase and test them in SQLite:

- a) Which stock has Sally ordered?
- b) What are the names of customers who live in Nathan?
- c) What are the names of customers who have bought Marcel's Morsels? Note: there is a small catch here where you need to deal with ' in the string.

For task 4, you need to execute your query and show your reviewers that your queries are producing the correct result.

### Task 5 – SQL version of PMs database

- Create an **sql** subdirectory.
- Copy the **pms.sql** file into the **sql** subdirectory.
- Study this file which contains a SQLite table definition and sample data for this example.
- Create a **pms.sqlite** database in the **sql** directory and **.read** the pms.sql file into it following the steps in task 3 and 4.
- Give SQL queries to check that the database contains the intended Prime Ministerial data. For example:  

```
select * from pms;
```
- Note that the names of some fields/columns have been changed (index → number, from → start and to → finish).