

Accessing an Elements Database

To do this exercise you will need to have installed a browser and editor for SQLite databases. See the instructions on Blackboard for installing either the *DB Browser for SQLite* standalone application or the *SQLite Manager* add-on for Firefox.

A. Restoring the elements database

Download and save the SQLite script `elements.sql` from Blackboard. This script creates and populates two tables, `atomic_numbers` and `atomic_symbols`, containing data about atomic elements. Create a database file `elements.db` and populate it by running the script. In the *DB Browser for SQLite* application this can be done in two separate steps, but an easier way is to choose the 'Import database from SQL file' option and follow the prompts.

(There are minor differences in different versions of the SQLite language. Just in case you have any problems running the SQLite script provided we have also supplied a copy of the `elements.db` file, but it would be better for you to run the script yourself to see what it contains.)

B. Examining the elements database

Browse the contents of the `atomic_numbers` and `atomic_symbols` tables to familiarise yourself with their contents. In the *DB Browser for SQLite* application this is done easily via the 'Browse Data' tab.

C. Extending the elements database

One of the tables matches elements to their abbreviation and the other matches elements to their atomic number.

1. The twenty-first element is Scandium, abbreviated 'Sc'. Use your database editor's interface to add this element to both tables. In the *DB Browser for SQLite* application this is done via the 'Browse Data' tab. Existing cells can be edited just by clicking on them and new rows can be created by pressing the 'New Record' button.
2. The twenty-second element is Titanium, abbreviated 'Ti'. This time, instead of using the editing interface, write and execute two SQL **insert** statements to add this information to the database. In the *DB Browser for SQLite* application this is done via the 'Execute SQL' tab.

In the *DB Browser for SQLite* application note that none of the changes you have made will be stored in the database file itself until you press the 'Write Changes' button. Until you do this you have the ability to undo all your changes by pressing 'Revert Changes'.

D. Querying the elements database

Now you can start performing queries on the two tables.

1. Write and execute an SQLite query which returns all element names with atomic numbers less than or equal to 10. (The answer should be the elements from Hydrogen to Neon.)

2. Write and execute an SQLite query which produces a table of element abbreviations and corresponding atomic numbers. To do so you will need to 'join' data from both of the tables. (The answer should link 'H' to 1, 'He' to 2, 'Li' to 3, and so on.)
3. Write and execute an SQLite query which returns the symbols for all elements with atomic numbers greater than 10. Again you will need to use the data in both tables to do this. (The answer should be the elements with symbols from 'Na' to 'Ti'.)