

Accessing an Elements Database

A. Restoring the elements database

Download and save the SQL script `elements.sql` from Blackboard. This script creates and populates two tables, `atomic_numbers` and `atomic_symbols`, containing data about atomic elements. Import and run this script using the *phpMyAdmin* interface, via the **Import** tab. Make sure your database is selected (highlighted) in the database list on the left before doing so, to indicate which database the script applies to. (You may only be able to see your own database in the list, depending on the privileges you have been granted.)

B. Examining the elements database

Use the *phpMyAdmin* interface to examine the contents of the `atomic_numbers` and `atomic_symbols` tables. Simply clicking on the table in the left-hand column will open a **Browse** tab which lets you see (and edit) the table's contents. Notice when you do so that this is actually achieved by running an SQL **select** query, which is shown up the top of the screen.

C. Extending the elements database

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

1. The twenty-first element is Scandium, abbreviated 'Sc'. Use the *phpMyAdmin* interface to add this element to both tables. First select the appropriate table in the list on the left. Then select the **Insert** tab and enter the values for the new row. When doing so, pay attention to the SQL **insert** statement generated to update the database.
2. The twenty-second element is Titanium, abbreviated 'Ti'. This time, instead of using the **Insert** interface, write and execute two SQL **insert** statements to add this information to the database using *phpMyAdmin*'s **SQL** tab.

D. Querying the elements database

Now you can start performing queries on the two tables via the **SQL** tab.

1. Write and execute an SQL query which returns all element names with atomic weights less than or equal to 10. (The answer should be the elements from Hydrogen to Neon.)
2. Write and execute an SQL 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 SQL query which returns the symbols for all elements with atomic weights 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'.)