

Theme: Semi-structured data, XML and RDF, Lab exercise part 1:

Objectives: The purpose of this lab is to create two data models and to reflect over their differences and similarities.

Your task is to create the data model for a web application which features experts and their expertise. The application contains a network of experts. For these experts we want to record information about their expertise and knowledge and which other experts they recommend. (The application is a combination of Wikipedia and LinkedIn).

Your application must at least contain the following information:

1. Experts are described by a unique identifier, a name, an e-mail address and *a semi-structured textual description about the expert*;
2. Expertise areas which are simply a set of topics. A piece of text is associated with each topic. A given topic can be subtopic of some other topic, for instance, SQL is a subtopic of databases which in turn is a subtopic of computer science. A given topic can be subtopic of more than one topic. Each topic has a single expert as its originator. (Hint: Consider topics the same as expertise areas);
3. Experts can be experts within a number of expertise areas. (Note: 'an expert has expertise areas' is different from 'an expert is an originator of expertise areas');
4. Experts can recommend other experts and any recommendation can be justified by a piece of text.

The application can be general or more tailored to a specific area, if you wish. You may also add more information to the model, but check with your lab assistant first so that your data model doesn't become too large.

1. Create a relational database for the application. Create the relations in a relational database engine and add some sample data;
2. How can the problem be modelled in XML? How can the different parts of the application be represented as XML? What alternatives are there (in principle), and what pros and cons do you see with the alternatives (e.g. elements vs attributes, representation of many-to-many relationships)? Choose one alternative and create an XML model for the above application using an XML schema. You have to make use of the *key/keyref* feature in XML at least once. (Recommendation: The XML model should not follow the relational model directly.);
3. Create an example XML file with data. Verify that your schema and data file validates;
4. Which parts of the above application were most suitable to represent as relations, or XML. Why?
5. How both models can be combined in this application? How the combined model can be stored? Which parts of the application would you create as relations, XML? Justify.

Make a report with your models and answers to the above questions. The report must include:

- an ER-diagram of the relational model (can be hand written);
- relations and content for the relational database (these are preferably the scripts you used to generate them);
- the XML schema for the data;
- the XML data file;
- answers to questions 2, 4 and 5.

The relational and XML models created here will be used in the next lab.

Please do not use the user interface provided by the tools and write everything manually.