Nicholas Marengo

Database Management

Prof. Labouseur

Lab 1

http://labouseur.com/courses/db/

Data vs. Information

One of the most popular (and arguably the oldest) form of a database today is a library or bookstore. The way that books are organized in such a place allows them easily be found and accessed by customers and the public. In this example, the elements of "data" include the many books, tapes, DVDs, and CDs that are available in the library. The "information" is how all of those products are organized in the facility: by using a category and alphabetical system, the products can be found very easily so that the "data" is organized and makes sense to the user. What good is a library if all of its contents are strewn about randomly, with no means to find what it is you're looking for?

By definition, data consists of unorganized facts that need to be processed in order to make sense. Information processed and organized data, or data that makes sense and it legitimately useful. Let's say someone were to head to the bookstore to look for Lucasfilm and Del Rey's newest paperback release, *Star Wars Maul: Lockdown*. They enter the store, but immediately trip over a pile of Metallica CDs and Isaac Asimov novels. They look around, and there are books and CDs seemingly piled randomly everywhere. There are no bookshelves, no tables, no signs or labels, and not even an aisle to walk in. This is what a bookstore would look like without having organized its "data" into "information".

Once all of the data is organized into a coherent manner, it then becomes "information". In this example, products in a bookstore are organized first by what it physically is (book, DVD, CD, other) and then often by genre before being placed alphabetically on the shelf. All in all, the data was structured, organized, and processed so that it makes sense to the user and has a useful context – and became information.

Data Models

The pre-relational hierarchal data model starts at one main node, and branches off into its next nodes. If one or more of those nodes contains others that are included, that node branches off as well. If one or more of the attributes are shared, it gets repeated. The network model is extremely similar in design, but it makes somewhat of an improvement because if two attributes share another attribute, that attribute gets connected to the parent nodes. Unlike the hierarchal model where that node is repeated, the network model condenses the model slightly more.

The relational model improves both previous models significantly because it takes each main node (the ones that branch immediately off of the first one) and lists all of their attributes in one place. It does this with each node and then connects them with branches. If a side of the branch has three prongs, it means that it is contained "inside" of the node it is attached too. This model condenses the nodes even more as well as makes them easier to display and comprehend. Considering this, I find XML as a decent model for data storage because, in a way, it seems to combine the relational and pre-relational models. XML is hierarchal with inherent ordering, but does not go into as much detail and is not as concise as the relational model.