The Importance of a Database

In today's information age, data is very valuable, and it is important to manage data properly. Data management can comprise of multiple steps such as:

- **Persisting data**: Data persistence means storing information properly so that it can be retrieved efficiently later on.
- **Organizing data**: Using proper data structures to store data so that disk space is utilized efficiently.
- **Manipulating data**: The data stored should be able to support `Create`, `Read`, `Update` and `Delete` (known as `CRUD` operations).
- **Searching data**: Making sure that data searches are performance-efficient and comprehensive.
- **Securing data**: Data is very valuable from a business standpoint. Proper security measures should be in place to make sure that data security is not compromised.

To achieve these goals, it is very important to select the correct data store, which is also called a database. The proper choice of a database depends on the application's actual requirements.

Selecting the Right Database

Most commercial applications today rely on some sort of external database for persistence. The most common methods to store data are:

- Text files
- XML files
- Relational databases
- Any other proprietary format

Text files are common for very small projects where we do not need to store complex data structures and there are no concurrency issues caused by the data being updated by multiple users. Adding the ability for users to simultaneously add or modify data in text files will take a lot of effort, and for such scenarios, it is recommended that you use a **Relational Database Management System** (**RDBMS**).

But if your application doesn't need to store much data, then there is no need to go for an RDBMS. We can simply use the text files for the rarely-used data. Some examples of such applications can be:

- File encryption utilities that simply encrypt data in memory and then decrypt it as required. These files might need to store the encryption key in some text (or custom file) format.
- Other simple desktop based programs such as a calculator, calendar, and so on.
- Simple websites that display static information (such as company information) and do not store any data.

XML files are a better alternative to text files as they are generic and can be processed by multiple systems. XML files are also easy to access programmatically as there are a lot of in-built libraries in the .NET framework that can handle XML files. XML files can also handle large amounts of data efficiently, and use XPath as a query language to fetch data. XPath is a simple language that can be used to select nodes within any XML document. XML can be a good choice for an application such as a blogging platform. Storing information such as blog entries in XML format is convenient for producing **RSS** (**Really Simple Syndication**) feeds.

RDBMS is one of the best and most commonly used database formats today. Most of the commercial RDBMSes, such as SQL Server, Oracle, and so on provide a lot of advanced functionality so that the developer does not need to bother about the standard data management issues. RDBMS is the best choice for most commercial applications that need flexible data storage as well as efficient data retrieval and processing.

When selecting between XML and RDBMS, we need to consider the following points:

- XML data can be easily distributed and understood by different systems since it is a standard markup language supported by wide variety of diverse platforms. Transferring RDBMS is not that easy, as each RDBMS typically has its own proprietary format.
- An RDBMS has a lot of built-in tools for manipulating and querying data easily, whereas XML lacks such tools. Moreover, one has to create or buy custom tools for XML files in order to be able to manipulate data as easily as a commercial RDBMS can do. So it is obvious that an RDBMS is more than just a data file; it is an advanced program that provides a rich interface to handle data, whereas XML is just a simple file that has the data stored in a predefined format.
- Querying XML data (using XPath and XQuery) is more complex than using **Structure Query Language** (**SQL**) statements, which are supported by all major RDBMSes. So if your application queries a lot of data it would be better to use an RDBMS.