Hammering Screws or Designing Databases

by Dennis Santoro

Although everyone knows you cannot build a house from the roof down I am constantly amazed at how many people try essentially that with database development. While it may seem that there is no similarity between building a house and building a database let me extend the analogy to illustrate what I mean.

Let us start with tools. Most people can be trained to use a hammer. Conceptually it is very simple. You swing a weight on a handle which amplifies the force that you can apply. This drives the object struck (a nail) into another object. So far so good, but anyone who has ever used a hammer knows that people's skills vary in areas such as aim and ability to apply power. Further, most people know that you do not hammer screws. Although screws do the same type of work as nails they tend to be used for different reasons and are applied with a special tool; a screwdriver.

Now return to databases for a minute. Think about how many people design "databases" in their spreadsheets because they happen to have database and query functions. Often this requires enormous amounts of redundancy, blank cells or both and requires incredibly convoluted formulas to do useful work with the data. These are the people who are banging screws with hammers or trying to drive nails with screwdrivers. They may get the job done but it is far from elegant and efficient. Often, over time, the cost of this misapplication of tools can be substantial in both staff time, lost opportunities and inability to access data in useful and necessary way. Now back to the analogy.

When some people begin to realize there are other tools they begin to learn about what is available and which ones are right for which job. They learn about framing hammers and finish hammers. They learn when to use what type of nail or screw. Then they learn about the variety of saws available and when to use each. Slowly they become a tool expert. People ask them advice about how to use tools. They know the "Tool Guru" can tell them which tool to use and exactly how to use it. Then the "Tool Guru", flushed with their own success, decides to take the next logical step. The "Tool Guru" decides to design and build them a house. The "Tool Guru" has seen houses before, lived in them, even fixed a thing or two when something broke so they make the mistake of thinking "sure, I can do that." Unfortunately, they usually cannot because they do not have all the skills and knowledge required; they are just great tool users.

Back to databases. Many people in organizations and, unfortunately, operating as consultants, are rightfully perceived as great tool users but accept the challenge of "building the house." Unfortunately, being able to code in dBASE, design amazing Paradox queries, develop slick Access forms or use any of the software available for database development does not qualify a person to design and build databases any more than our "Tool Guru" is qualified to design and build a house.

If the "Tool Guru" correctly understands his strengths and limitations he either brings in an architect or acquires the skills of one. The architect will begin with such basics as asking you about what you want and need. How many people will live in the house? Do you like open floor plans? How much growth space do you anticipate needing? What styles, colors and materials do you like? What does the building site look like? How much do you want to spend? What features are mandatory and which are optional? Then they will work up plans which they will review with you and revise them until they have a design that meets your requirements and preferences. These plans can then be used by someone who knows how to read blueprints, in conjunction with someone who knows how to use tools, to build the home as desired.

Although this process should also be fundamental with databases, quite often it does not happen at all. This is especially true in smaller organizations where there is little or no formal IT structure or the IT resource is devoted to the infrastructure but not the system development effort. This is why many database projects fail and why many others are ramshackle at best. The lack of a good design process may be understandable in light of the still prevalent perception that computers operate by some kind of magic which the average person will never understand. This does not make it any more appropriate as a development strategy. In addition, the relatively low cost and prevalence of the tools often mislead people into thinking anyone can master them or worse, no mastery is required. Unfortunately that is not even true for hammers, let alone today's most common software. This is also fostered by the way many vendors hype their "Experts" and "Wizards" as able to do the work for you. This is the same fallacy which many purchasers of project management software discovered in the late 80's and early 90's. While the technology is quite useful if you know what you are trying to accomplish and how it should be done it is useless or worse if you do not have the proper background information.

A good database design process will involve the assessment of the needs to be addressed, the modeling of data relationships and business rules, appropriate structuring of data sets (normalization) and design reviews with users before a single line of code is written. As the database development proceeds there will be further review, usability tests and communication between the designers and the users to ensure that the final product actually meets the organization's needs. Generally, the investment in good design will provide significant payback in functionality, ease of use, user satisfaction and improved productivity. Often it can make the difference between an organization's growth and success and its demise.

Don't let the tools fool you. Architectural skills are still important whether building a house or a database.

Conditions of Use and Reproduction:

This content in this paper is provided as is, with no warranties, guarantees, or claims regarding its accuracy, completeness, or usefulness. While all efforts have been made to ensure its quality and accuracy, you are solely responsible for your own use of this information.

Any statements of fact contained in this article should be interpreted as the opinions of the author, which may or may not reflect the opinions of any other entity involved in the transactions that led you to this article.

Feedback is strongly encouraged, especially constructive criticism and/or typographical/syntactical corrections. Response or action is left to the discretion of the author. Flames, abusive language, unsolicited commercial email messages (aka SPAM), and other forms of rudeness will be cheerfully ignored. Professional responses will receive priority attention. Unprofessional contact will be ignored.