

## Is Your Business Normalized?

by Kevin Lee Elder

Biography: Kevin Lee Elder is an associate professor of information systems at Georgia Southern University. He may be reached by e-mail at [kelder@georgiasouthern.edu](mailto:kelder@georgiasouthern.edu).



Most organizations are plagued with information systems problems. They have the data, somewhere in their systems, but getting to the data, or presenting the data in a usable format is too difficult. There is a simple reason for most of this. The systems most of us use on a daily basis we're not designed for their present usage. That is, they have grown to their current state from many different and smaller systems we used to have. What is the solution to this problem? It's simple, we need to design our systems, or redesign our systems, in order to make them meet our needs for today.

There are three main aspects to a good system design. First there is the process design. In this design we determine where the input to the systems will come from and how will we capture the input data. Then what processing needs to be performed on this input. Finally we ask what output do we need, just what views of this system are required or desired. We normally use data flow diagrams to model these processes and occasionally the existing systems are in such a bad state that rather than a simple redesign we are forced to do a business process reengineering (BPR) of the system. In BPR we break the current system down into the basic processes and try to determine an entirely new way to put the processes together to design a new system. In this design we use the various techniques described in a previous column to determine what the real problems are with the existing system in order to design a sound solution.

The second type of design is the user interface (UI) design. In this design we determine what the user views of the system need to be and we design the dialogue and methods that the users would like to use to interact with the system. This will entail designing menus, forms, reports and other user interfaces. In this design we often use Web Usability processes to determine the best methods to design our interfaces as we have discussed here in a previous column.

The third type of design is data design. In this design we take all the input and output flows identified in the process design, that we further identified and modeled in our user interface design, and now we determine how to store this data in a relational data model. Most of the systems we use today store the data in some form of a relational database so we need to use methods and techniques to design what that relational model will look like. This is where we can really try and solve those issues and instances where the users know they have the data, somewhere in their systems, but getting to it is usually too difficult. The best tool and technique I know of to design the relational model is something called normalization.

In normalization we start with the user views (the input or output process) and go through a series of structured, repeatable steps to break the data down into a design that will eliminate redundancy and allow easy access to each and every piece of data uniquely in the system. If we use a structured repeatable and easily understood process to design each of our user views we should be able to combine them all into one big enterprise-wide design. This normalized data design will allow access to all the data in all the systems we use. If we leave the data un-normalized the data will not be as easily accessed, processed or viewed, which is where most systems are today.

In normalization there are a series of steps we follow to break the data down into tables, unique keys to access the tables, and indexes to interrelate the data/tables. Are you limited in the number of telephone numbers you can store for a customer? Or the number of products you can store for an order? If so your data is not in first normal form. If you cannot add or delete some data without losing other pieces of data you are not in second normal form. In second normal form we eliminate the need to enter dummy orders, customers, or products (number 999999 or date 1999) so we can add data into the system. If you have to know the invoice number you shipped the products out to the customer with in order to get to the data from when the order was placed, you probably are not in third normal form. Most normalized systems are at least in third normal form, which gives us all the views of the data in the system today and most of the views we will need in the future.

If we follow all the steps in the normalization process to the very end we can uniquely find every single piece of data in the system, provided we have plenty of processor speed and lots of data storage. Baseball is the best example I can think of to

illustrate who might want complete normalization. If we had batting averages stored this way, no matter what situation arose in a game: runners on second and third, bottom of the seventh inning, left handed relief pitcher, second game of a double header, behind by one run, Astroturf field, wind out of the west 5 MPH . . . Baseball runs into almost unique situations each game, each inning. If the right manager could search and easily find the right information, at the right time, instantly, they need this type of normalization.

So in my consulting work, and in the classroom, we always normalize our data so we can access it later. We usually stop at third normal form. Is your business struggling to access your data, or are you normalized?