Craig S. Mullins & Associates, Inc.

Database Performance Management

Return to Home Page

March 1996

Dealing With Data Outhouses

by Craig S. Mullins

There is quite a bit of discussion these days about moving to a data warehouse environment. Terminology is thrown about pertaining to data transportation, data replication, and data propagation, but little thought is put into data purification. Oh, sure, we hear about data scrubbing and data cleansing, but the true scope of the problem is rarely defined. And that scope is immense.

The premise of this article is that most of us are stuck with data outhouses instead of data warehouses. Much of our data is dirty, and we don't even want to consider what it would take to clean it up. The age old adage "garbage in, garbage out" still applies and there is nothing we can do about it short of analyzing and correcting our corporate data. Failure to do so will result in poorly-made business decisions.

Defining the Scope of Data Outhouses

We've all had that experience where we look at the contents of one of our major flat files or database structures and intuitively know that the data is incorrect. There is just no way that that employee was born in 1989. You know your company doesn't hire six year olds (even if some of your co-workers seem to act that age)! And that next record looks bad, too. How could she have been born in 1978 but hired in 1977. Most companies don't hire unborn embryos.

All too often, these types of data integrity problems are glossed over. "No one would actually take that information seriously, would they?" Well, maybe people won't, but computerized systems will. That information can be summarized, aggregated, and/or manipulated in some way, and then populated into another data element. And when that data element is moved into the data warehouse, analytical processing will be performed on it that can impact the way your company does business. What if warehouse data is being analyzed to overhaul hiring practices? It may make an impact on the business decisions if enough of those hire and birth dates are inaccurate.

Small data discrepancies can become statistically irrelevant when large volumes of data are averaged. But averaging is not the only analytical function that is employed by analytical data warehouse queries. What about sums, medians, max/min, and other aggregate and scalar functions? Even further, can you actually prove that the scope of your data problems is as small as you think it is? The answer is probably "no."

And this is just one small example of the scope of the data integrity violations that many application systems allow to be inserted into production data stores. Some of the integrity

violations may seem to be inexcusable. For example, most of us have experienced the GENDER column (or field) that is supposed to store “M” or “F”. Frequently, GENDER data can be seen that defies imagination—everything from “*” to “!” to a blank. These designations typically do not refer to hermaphrodites and eunuchs; they are incorrect data values. Shouldn’t it be a simple matter to programmatically force the values to be either “M” or “F”? The short answer is “yes,” but this simplifies that matter too much. Many systems were designed to record this information, if available, but not to force the user to enter it. If you are a telephone marketer, the reasons for this are clear. Not everyone wants to reveal personal information and it is not always an easy matter to independently acquire the information. However, the organization would rather record incomplete information than no information.

The organization is correct in desiring incomplete information over nothing. However, there is still an ignored problem. The true problem is that a systematic manner of recording “unknown” values was not employed. Every program that can modify data should be forced to record a special “unknown” indicator if a data value is not readily available at the time of data entry. Most relational DBMS products enable data columns to store a “null” indicating “unknown” or “unavailable” information. Pre-relational DBMS products and flat files do not have this option. However, some specific, standard default value can be chosen. The trick is to *standardize* on the default value.

Cleaning up the Data Outhouse

Currently, there is no way to completely avoid human interaction when attempting to clean a data outhouse. Data scrubbing is a common term for cleaning up data as it is moved into the data warehouse. This usually refers to changing codes into meaningful values. For example, a CUSTOMER-CODE of 5 means nothing to the typical user. But a CUSTOMER-CODE of "Corporation" or "Individual" is usable and helpful.

This type of processing should be the second pass at cleaning out the data outhouse. The first pass should be the standardization of "unknown" values. This can be a tedious process. Our primitive examples in the previous section utilized data elements with a domain of 2 valid values. Most data elements have domains that are considerably more complex. Determining which are valid values and which are not can be difficult for someone who is not intimately aware of the workings of the application systems that allowed the values to be inserted in the first place. Is '1895-01-01' a valid date for that field or is it a default for an "unknown" value?

Only an in-depth analysis of the programs and the meta data in the corporate repository can provide the answer. 19th century dates may be valid for birth dates, stock issuance dates, account inception dates, publication date, and any number of other dates with long periods of "freshness." And, just because the program allows it to be put there, that does not mean it is actually a valid date! It is quite simple for a user to type in 1895 instead of 1995. If the data entry program is not intelligent enough to trap these types of errors, your systems will insert dirty data into production data stores. This type of data integrity problem is the most difficult to spot. It is quite likely that only the business person who most uses the data can spot these types of problems.

A Light at the End of the Outhouse

So what is the solution? Several techniques can be used, but the best approach is to foster an environment in which data is truly treated as a corporate asset. I know, I know, you've been hearing this for years. But that doesn't make it any less true. The problem is attracting the appropriate high-level management personnel who can implement a policy that values data.

What does this mean? Consider the other assets of your organization. The capital assets ($) are modeled using a chart of accounts. Human resources (personnel) are modeled using management structures, reporting hierarchies, and personnel files. From building blueprints to item bills of material, every asset that is truly treated as an asset is modeled. If your corporation does not model data, it does not treat data as an asset and is at a disadvantage.

That said, if your corporation does have a data model, the task of cleaning up the data outhouse is simplified. At least you know what the valid values should be. Of course, you still have to do the physical clean-up. Automated tools exist that can help you with this, but they can not do it all for you, yet.

Of all the automated solutions available, repository technology can be one of the most helpful if utilized properly. A correctly implemented repository will house the meta data and the data model for the corporation. It can act a single, centralized store to assist in the migration of data from the outhouse to the warehouse.

Alas, many shops do not own a repository. Even worse, some of those that do own a repository, neglect the product causing it to become "shelfware." There it sits on the shelf and the meta data in the product is either outdated, inaccurate, or non-existent. This does not negate the value of repository products, it simply depicts the cavalier attitude that many organizations take toward their data. If you own a repository, the single most important thing that you can do to enhance the value of your data is to keep the meta data in the repository up to date. This requires a lot of effort, a budget, and most of all, commitment.

From the Outhouse to the Warehouse

Awareness of the problem is the first step. But what if you know that you have a data outhouse and want to clean it up? What follows are the top ten things you can do to begin the move from the data outhouse to the data warehouse:

1. Foster an understanding for the value of data and information within the organization. This can be accomplished through lobbying the users and managers you know, starting an internal newsletter, circulating relevant articles and books throughout your company, and treating data as a corporate asset yourself. A lot of salesmanship, patience, politics, and good luck will be required, so be prepared.

2. Never cover up data integrity problems. Document them and bring them to the attention of your manager and the users who rely on the data. It is usually the business units using the data who are empowered to make changes to it.

3. Do not under estimate the amount of time and effort that will be required to clean up dirty data. Understand the scope of the problem and the process required to rectify it. Take into account the politics of your organization and the automated tools that are available. The more political the battle, the longer the task will take. The fewer tools available, the longer the task will be. And, even if you have tools, if no one understands them properly, it will probably be worse than having no tools at all as people struggle to use what they do not understand.

4. Understand what is meant by a data warehouse. A good definition of data warehouse is provided by the book, ***Essential Client/Server Survival Guide***:[1] "a data warehouse is an active intelligent store of data that can manage information from many sources, distribute it where needed, and activate business policies." Other defining characteristics of a data warehouse are:

    - it is read only
    - it is separate from production, transaction data stores
    - it typically contains a vast amount of data whereas production data stores usually undergo periodic archival
    - the data is formatted for retrieval.

5. Educate those implementing the data warehouse by sending them to courses, industry conferences, purchasing

books, and reading periodicals. A lack of education has killed many potentially rewarding projects.

6. Physically design the data stores for the data warehouse differently than the similar, corresponding production data stores. For example, file and table structures, indexes, and clustering sequence should be different in the warehouse because the data access requirements are different.

7. It is often stated that denormalization is desirable in the data warehouse environment, but proceed with caution. Since denormalized data is optimized for data access and the data warehouse is “read only”, it would seem that denormalization is a natural for this environment. However, the data must be populated into the data warehouse at some point. Denormalized data is still difficult to maintain and should be avoided if performance is acceptable.

8. Understand the enabling technologies for data warehousing. Replication and propagation are different technologies with different availability and performance impacts on both the production (OLTP) and the warehouse (OLAP) systems.

9. Only after you understand the basics should you delve into the more complex aspects of data warehousing such as star schema and multi-dimensional databases.

10. Reread steps 1 through 9 whenever you think you are over-worked, underpaid, or both!

Synopsis

The need to create and maintain a data warehouse is becoming a business reality. But, as IT professionals, we must understand that the data in the warehouse is only as good as the sources from which it was gleaned. Failure to clean dirty data can result in the creation of a data outhouse instead of a data warehouse.

From *DM Review*, March 1996.

---

[1] Essential Client/Server Survival Guide, by Robert Orfali, Dan Harkey, and Jeri Edwards. Published by Van Nostrand Reinhold, New York, NY, 1994.

Home. Phone: 281-494-6153 Fax: 281-491-0637