Delivering information you can trust
April 2008

IBM Information Management software

The case for data profiling

By David Plotkin

Contents

The case for data profiling

Organizations who are serious about improving the ability to deliver critical business initiatives and ensuring that the quality of their data supports those initiatives are turning more and more to a rigorous, repeatable process for analyzing the quality of their data and taking steps to not only improve the quality of that data, but to protect that quality going forward. This effort involves not only understanding where the data quality issues exist, but also the root cause of the issues and which issues cause enough "business pain" to make them worthwhile to fix. This paper discusses ways to understand and fix data quality using "data profiling".

Reactive versus proactive data quality

You can often tell a lot about the maturity of a data-driven organization by how that organization approaches data quality. Organizations that have just begun to realize that data quality is important typically engage in "reactive data quality". This type of approach is characterized by a tendency to only react to bad data quality after it is discovered (usually by the end-user), and to not have a clear and well-documented idea of what constitutes quality data – that is, a lack of any documented data quality rules. In addition, data quality issues are not systematically recorded, nor is there a clear path for reporting and escalating data quality issues that are discovered. Where a data quality practice is either missing or strictly in reactive mode, it is not uncommon for an organization to spend 50% or more of its analysis time attempting to ferret out information from inaccurate or incomplete streams of data. In fact, this author is aware of an organization in which the stored data in the data warehouse was considered so untrustworthy that an entire group of people (and their tools) was dedicated to extracting and analyzing that data – and applying a whole series of data quality rules in an attempt to figure out what data was good enough to use.

More advanced organizations tend to lean toward a much more proactive approach. They understand the value of good quality data, how to discover the specifications for determining whether data is of good quality, and how to use those data quality rules to inspect the data. Further, they can even enforce those data quality rules during the data warehouse load, so data quality issues are known BEFORE they impact the business. "Trapping" bad data in special reporting environments makes it possible to respond quickly to perceived data quality issues. This early discovery of data quality problems increases the customer's confidence in the data, leads to better business decisions, and eliminates a lot of the rework and frustrations of having to "massage" bad data In addition, a robust data quality process is often less expensive to maintain than all the analysis and cleanup that inevitably goes with poor-quality data.

Key components for successful proactive data quality

There are certain key components necessary for performing proactive data quality, including a strong involvement from the business, support from IT, a way to inspect large volumes of data (called data profiling), and strong protocols for documenting and publishing the information gained while working with the business people and performing the data inspections.

Strong involvement from the business

The strong involvement from the business comes by way of "data stewardship". The data stewards are a group of business people who either know the data themselves, or have access to people who DO know the data (and preferably, work with it day to day). The data stewardship committee is typically made up of representatives from the various functional areas of the business, such as finance, marketing, risk, and operations. Their job is to provide formalized accountability for key data-related information (metadata) such as definitions, derivations, data quality rules/requirements, and where the data originates (which could be in multiple places). All of this information is crucial to a successful data quality

initiative. After all, how can you "do" data quality if you can't agree (or have someone specify) what a data element means (the definition), how it is calculated (the derivation), when it is considered to be of good quality (the data quality rule/requirement) and what constitutes the system of record for that data. Further, the committee must determine and agree upon which functional area has accountability for specific data elements, and which data elements are worth dealing with, since there are usually many more data elements than can be handled all at once.

Strong support from IT

IT support is necessary because figuring out what a data definition means and how the element is derived often means diving into the depths of aging and poorly documented applications. IT can help you do that, and often has the requisite knowledge of how things work because they have had to maintain the applications and revise them as business requirements change. In addition, if the data stewards decide to actively enforce the data quality rules during a load, IT will need to build (or revise existing) extract, transform, and load (ETL) jobs. Other tools – such as a suite of data profiling applications and a metadata repository – need IT support as well. Note that since IT is a service organization (or should be), it is up to the business to prioritize that resource to perform the data quality-related work.

Gathering and using data quality Metadata

Strong protocols (and the right tool support) for recording data quality-related metadata is another key aspect of a successful data quality effort. A proactive data quality effort (and even a reactive one) exposes a lot of really important information. In addition to the definitions, derivations, and data quality rules/requirements mentioned previously, the results of large-scale data inspection and the business feedback on those results needs to be recorded. For example, if a data quality defect is detected, documenting that defect and the resolution (including deciding whether or not it is worth fixing) will alert future users of that data to the issue, and enable them to make a more informed determination as to whether to use the data or not.

Metadata about where the data originated and what was done to it on its travels (known as "lineage") is crucially important as well. Understanding where the data "stops over" in data stores simplifies the data quality effort because the definitions, derivations, and rules/requirements that apply to the data in one data store should also apply to that same data when it is stored somewhere else. Or, if the data has been manipulated (such as converting one set of valid values to another), understanding how that manipulation was done lets you determine the data quality rules/requirements much more easily than starting from scratch.

Most companies today have migrated to strong Extract, Transform, and Load (ETL) tools, which keep metadata about lineage. Developers who build the ETL processes can get significant benefit from the results of data profiling, as they will then know what to expect from the stream of data that the ETL job will use. Most ETL developers admit that they do data inspection (essentially, a very simplified form of data profiling) anyway to ensure that their code will run. But the results of that data inspection are rarely published to others who could use the information, or formally recorded for future use. A formal data profiling program can save all this effort, provide more robust results, and ensure that the people who need the information do, in fact, get it.

In addition, modern data profiling tools also generate a lot of metadata about the condition of the data and how well it meets the specified data quality rules/requirements. The data profiling toolsets often integrate the profiling metadata with the ETL metadata (at least, if they are from the same vendor), enabling the ETL tools (sometimes with an assist from the Data Profiling tools) to enforce data quality during the load of the target database, and enabling the ETL developers to leverage the profiling discoveries.

Recording data quality Metadata

All of this metadata needs to be stored in a "metadata system of record" – a robust metadata repository where it can be used and leveraged by analysts and other users of the data. Making the metadata accessible is important; siloed metadata is just as bad as siloed data. Thus, not only should the specific metadata related to data quality be recorded and reported, but the lineage from the ETL tools should tie in as well. Too often, the information about data quality gathered by a project is simply archived, never to be heard from again.

Data quality rules

So, what ARE data quality rules? Basically, they are the answer to the question: "when data goes bad, how do you know?" People who use the data day to day, and who recognize that something isn't right with the data are, in fact, applying these data quality rules, sometimes without quite knowing it. Some simple examples include:

- *This field is mandatory – I must have a value here (customer identifier)*
- *This field should have only certain valid values (there are only four types of loans)*
- *The data in this field must conform to a certain pattern of numbers and letters (Social Security Number or phone number)*
- *This field should have a certain data type and range (a FICO score is an integer between 300 and 850)*

Data quality rules/requirements can also be more complex, for example:

- *There is a relationship between one or more records (if the loan is a real estate loan, there must be information on the collateral for that loan)*
- *There is a relationship between one or more pieces of data (a field has 20 valid values, but only 3 of them are valid if another field contains "x").*

There are a number of ways to collect and validate data quality rules. The simplest occurs when a business person states the data quality rule for you, perhaps while discussing perceived data quality issues. Another way is to use a tool to ferret potential data quality rules out of the data. Either way, though, the data quality rule must be validated by the data stewards (one of their most important duties) and validated against the actual data. Using a tool to discover potential data quality rules, or validating stated data quality rules against the data are both tasks that fall under a practice known as data profiling.

What is data profiling?

Data profiling is the analytical process by which you examine the contents of a database and collect statistics and information about that data. The idea is to discover the structure, content, and quality of the data – and to do this whenever data is being converted, migrated, warehoused or mined. Note that Data profiling is defined as a process, and not a tool. The data profiling process breaks down into the following steps:

1. *Identify the business requirements for data quality and business data candidates for data profiling.*
2. *Identify required source systems, as well as files, tables, and data elements to be included.*
3. *Transfer the data into an environment where large-scale data examination can be carried out without impacting the transaction system or data warehouse response times. Under some circumstances, this step can be skipped for sufficiently robust servers and tools that support parallel processing.*
4. *Examine the contents of the database to see what is in there.*
5. *Construct a possible set of data quality rules to compare the data against. These data quality rules may be postulated by sophisticated profiling tools or proposed by business people who are familiar with the data (or both). If the data quality rules are postulated by the profiling tools, check these data quality rules with business people to find out if they are actual data quality rules, or simply a coincidence.*
6. *Compare the data quality rules against the data to determine how well the proposed data quality rules match the data.*

7. *Check the results with business people who know the data.*
8. *Present the findings to the business community.*
9. *Gather requirements on how to use the results to improve the data and ETL process.*

As you can probably guess, this is an iterative process (at least from steps 4 to 7). If the fit between the stated data quality rules and the data itself is not good, this may lead to a refinement of the data quality rule as the business people remember oddities about the data after viewing the results. For example, a business person might state that a field should be an integer between 300 and 850, yet the data contains a significant number of values of 999. Faced with this result, the business person might then recall that 999 means that the value is actually unknown. On the other hand, a poor fit between the data quality rules and the data might simply indicate that the data is of poor quality! Either way, you now know more about your data than you did before.

The benefits of data profiling

There are significant benefits to performing data profiling. By verifying the availability of information and validating the quality of that information, you can improve the predictability of project timelines and lower the risk of design changes late in the project. It can also save considerable time in both the coding and Quality Assurance stages of the project. Although they may not talk about it, most programmers (and ETL developers, as discussed previously) will examine the source data to ensure that their code will run without incident. However, they will not do it as efficiently or as consistently as a Data Quality analyst using the data profiling process. Having a robust, repeatable process that is supported by both tools and methodology will inevitably lead to higher-quality ETL jobs. Further, since the programmer's main goal is to get their code to run, they may not report issues or document their findings. The Quality Assurance staff gains considerable leverage from a data profiling effort as well. If the QA staff knows about data quality issues ahead of time, they can write more robust test cases and they also don't have to spend time troubleshooting why some of their test cases failed. Further, by profiling the resulting data in the target system, they can spot individual errors or error patterns during the QA cycle.
Other benefits include:

- *Supporting compliance and audit requirements*
- *Rapid assessment of which fields are consistently populated as expected*
- *Focusing of data quality efforts where they are really needed*
- *Improving visibility to quality data that supports business decision making*
- *Identifying transformations for data migration and integration*

Another benefit of data profiling is that it provides a lot of valuable information about the data itself ("metadata").
This information includes:

- *Understanding what is in use, and what is not*
- *Whether the assigned business name is appropriate to the data*
- *Whether the assigned description is appropriate to the data. If there is no description, understanding the data may lead to a description.*
- *What the data quality business rules/requirements are*
- *How well the data matches the data quality rules/requirements*

Data profiling types

There are three main types of data profiling activities: Discovery of "new" data quality rules, assertion testing of stated data quality rules, and visual inspection.

Data profiling discovery

One of the more amazing capabilities of modern data profiling tools is the ability to inspect huge volumes of data in database and files and look for potential data quality rules and relationships in the data. These data quality rules and relationships may well have been lost in the mists of time, and represent information that the data analysts are not aware of. As you can imagine, such an effort – often involving hundreds of tables, thousands of columns, and millions of rows – requires considerable processing horsepower and tools that take advantage of multiple processors and threads. Most large-scale installations replicate the data into another environment to avoid impacting the data warehouse or transaction server, but this actually makes it more difficult to do data profiling because of all the overhead required for replication. Thus, a prime consideration when evaluating a profiling tool is whether it (and the supporting repository) will scale as your data store increases in size.

This discovery process uses sophisticated algorithms to detect the data quality rules and relationships, and presents the results along with violations. Results can include patterns, actual ranges of values, lists of valid values, number of nulls, incidence of repeated values, records that have a relationship, and more. For example, the tool might detect that a field contains only four valid values, except for a very small percentage of the time. It would present the list of valid values, the distribution of those values, and information on the number of times a different value was found – as well as a sample of the data records where a different value is found. It bears repeating that this extensive collection of metadata should be stored in a metadata repository for use (and reuse!) by business users, ETL developers, programmers, and others.

Once the tool has reported on what it "thinks" the data quality rules are, it is up to the data quality analyst (or someone in that role) to review those data quality rules with knowledgeable business people to find out which are truly data quality rules, and which ones are simply data anomalies. The tool may also identify that there is a data quality rule, but get the rule wrong. A common example is misidentifying a narrow range of numbers as being (instead) a set of valid values of those numbers.

Using the discovery process is a good starting point when there is little existing knowledge about the data quality rules. This is because the tool makes use of inexpensive processor time to "brute force" large volumes of data without the expensive overhead of analysts that is required for assertion testing (discussed next).

Data profiling assertion testing

With Assertion Testing, data quality rules are stated prior to examining and profiling the data in the database. The data quality rules typically come from three sources: Business analysts, documentation, and program logic.
Data quality rules collected from business analysts typically arrive one of two ways – via complaints about perceived data quality issues, or as issues raised during a project. The complaints (which will often come to a data quality help desk or are raised during a meeting) are fairly straightforward to handle. A data

quality analyst must question the business analyst via a "guided interview" to ascertain what the data quality rule is, as well as how important the business analyst considers the perceived violation of the data quality rule. This data quality rule should then be validated by the data steward for that element, and if the data steward agrees, the data quality rule would then be documented for use in the Assertion testing process. The process for collecting the data quality rule during a project is similar, except that the issues are raised during project meetings and evaluated by the project manager (along with the data steward) to determine if the data quality rule is worth investigating.

It is sad but true that oftentimes very little is known about a system (and the data generated by the system) by the people who use the system. In that case, the data quality rules (and probably a lot of other types of business rules as well) may be determined by checking the documentation – if it exists. This documentation may have been supplied by the original vendor, or be part of the documentation for projects that customized the application. As with data quality rules gathered from analysts, the data quality rules have to be evaluated against business requirements to determine if the data quality rule is worth looking into.

Finally, the actual program code can be checked (if you have access to it) to see what data quality rules are enforced there. This effort can be daunting, and requires both a programmer (to figure out what the code does) and a business analyst to evaluate the results. Oddly, however, program code has an advantage over both talking to business analysts and reading documentation – it is always correct – at least, when it comes to describing what data quality rules the system is enforcing at the current time. That is, no matter what the analyst THINKS is happening, and no matter what the documentation SAYS is happening, it is the executing code that is the final authority on what is ACTUALLY happening. Further it is the running code which controls the actual data content that is produced, and is thus the final authority on that, as well.

Once the data quality analyst has gathered the data quality rules from various sources and evaluated those rules to see if they are important enough to pursue, the data quality rules are "programmed" into an Assertion Testing data profiling tool. How this is done depends on the tool, but most of the tools have an option to allow the data quality rule to be entered and evaluated without having to know specialized languages, such as Structured Query Language (SQL).

The last step is to run the tool, and evaluate the results. The tool output indicates how well the data quality rule matched the data – as well as a sampling of data quality rule violations. Of course, if the results indicate a poor fit between the stated data quality rules and the data, the data quality rules need to be reviewed to see if the problem is a poorly stated data quality rule or poor quality data.

Data profiling visual inspection

The last type of data profiling involves visual inspection of the data in the database. This technique finds data inaccuracies that are not easily formulated as boundaries, limits, or data quality rules. Inspection helps to find problems with business-related quantities, such as:

- *Frequency distribution of values*
- *Sums and totals of values*
- *Comparing counts or totals from one data source to another, especially multiple systems that source the same data (or data that the business THINKS should be same.*
- *Values with counts less than some interesting threshold*
- *"Interesting" text strings, such as "Don't know", "Not Provided", "Company A", or "999-99-9999"*

The impacts on Metadata

Besides improved quality, one of the key deliverables from a data profiling effort is metadata: information about the data and the data quality rules that apply to the data.

There are two kinds of metadata related to data quality: Data quality rule metadata and Data quality results metadata. The data quality rule metadata must record the data quality rule itself, the "authority" for the data quality rule (such as an individual steward, documentation, or program code), and the data element(s) that the data quality rule is attached to. Of course, the statement of the data quality rule can be quite complex, and may require a database just to record it. But once you HAVE recorded the data quality rule, anyone can go and look it up, saving a lot of time on future projects or when troubleshooting data quality issues.

The data quality results metadata includes a link to the data quality rule being tested (which is in turn is linked to the data for that data quality rule), as well as the details of the test. These details include how and when it was performed as well as a measure of how well the stated data quality rule matched the data. Profiling results in a determination that either:

- *The interpretation of the data given by the metadata is correct and the data is wrong, or…*
- *The data is correct and the metadata (data quality rules) are wrong…*
- *Unless they are both wrong (which also happens)*

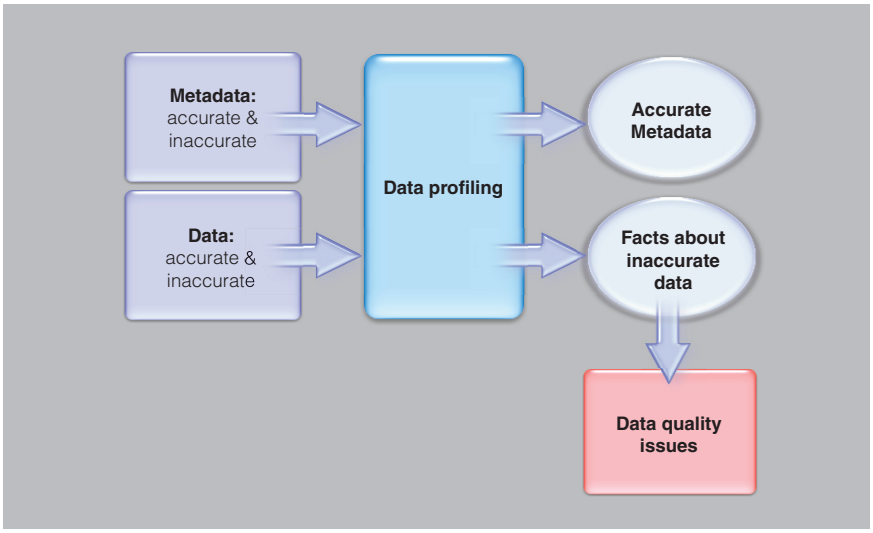

Figure 1: Architectural overview of data profiling

With so much information coming out of data profiling, it is imperative to have a "system of record" to record the data quality rules and results metadata. This is some sort of database (perhaps a metadata repository) where analysts can go to check current data quality rules, currently known data quality issues, and the results of data profiling including annotations from prior analysis.

It's all in the details: the four types of data profiling analysis

Data Profiling is all about rigor – performing the correct tasks in the right order to obtain the most useful results. There are four types of data profiling analysis: column properties analysis, structural analysis, data relationship analysis, and domain analysis.

Understanding column property analysis

Column Property analysis analyzes individual columns (fields) to understand all the values that appear in the column. That is, it looks at the values stored in a column independent of other columns. The type of profiling is both the easiest to do and the easiest for the business to understand. In addition, the data quality rules are the simplest. The output provides data on the values, frequency, uniqueness, patterns, data type, size and length, ranges, and maximum and minimum values.

The results of column property analysis enable analysts to:

- *Discover the metadata and content quality problems*
- *Validate that data conforms to expectations*
- *Compare actual data to target requirements, and*
- *Analyze invalid, default, and missing values*

Comparing the actual data to target requirements is a big win for projects. As mentioned earlier, programmers tend to perform this step on their own during code development to avoid nasty surprises in the data. If this is done (and documented) prior to the start of development, this individual (and typically inefficient) effort by the programmers can be avoided.

Understanding structural analysis

Structural Analysis focuses on the quality of entire data structures – that is, how various types of data (such as patient and prescription, or customer and order, or borrower and account) are tied together through relationships implemented in the database. It looks at defining identifiers (such as customer id) to evaluate whether these unique attributes are truly unique. It also ensures that data which should not exist without a relationship does not, in fact, exist in the absence of that relationship. For example, this type of analysis will detect whether an account exists without a customer. And depending on how your company defines "customer", it could detect incidences where the customer should not exist because there is no account (or prescription, or order, etc.).

Structural analysis is not particularly difficult to do, but it tends to be more work to evaluate the results. The reason for this is that, unlike column properties analysis (where data quality rules apply to just a single column and agreement is usually possible), structural analysis requires that a business come to an agreement on the definition and relationships between data structures that are shared across the enterprise. It should come as no surprise that agreeing on whether a customer can only exist if they have an account (or whether an account can exist without a customer) is much harder than agreeing on the valid range of values for a single column.

Understanding data relationship analysis

Data relationship analysis tests whether data relationships are correct. The first type (simple) tests the relationships within a single record or business object to ensure that there are acceptable combinations of values. For example, a customer record might have a field for the customer's name, as well as a type code which can take the values business or personal. The data quality rule might state that if the customer type was "personal", the customer's name was mandatory. But if the customer type was "business", then the customer's name was optional.

The second type of data relationship (complex) requires that data values conform to the relationship over multiple tables or business objects. This enables you to find prohibited conditions, such as that a customer cannot be both a retail customer and a wholesale customer. It also enables finding of missing or erroneous records, such as the fact that a home equity loan (recorded in the loan type) must have a matching collateral record. You'll want to look for a tool that allows cross-comparison of many domains (because the relationships often span domains) without requiring technical configuration or programming and utilizes previously discovered data content to rapidly uncover data quality issues. Note also that the relationships can be established across multiple source systems, thus aiding in system integration.

Understanding domain analysis

Domain analysis looks at values to try and discover aggregations, counts, or frequencies that appear to be unreasonable. Domain analysis enables the business analyst to spot things like:

- *Significant changes in volumes (25% fewer loan applications than last month)*
- *Significant changes in value (loan portfolio value jumped 40% since last quarter)*
- *Find "odd" relationships (loan-to-phone number ratio is 1.02 to 1, but several phone numbers show 25 to 1 ratios)*

Using the data profiling results to advantage

Data profiling leads to a concise, documented, and validated set of data quality rules and requirements. These data quality rules can be used to detect errors in the data BEFORE the data is loaded into a database (such as a data warehouse), or merged into a consolidated system (as in a data integration effort). The data quality rules can be applied when the data is moved from the source to the destination and corrective action taken during this load process to call attention to the errors – or even prevent the bad data from being loaded. To handle the detection of data quality errors during the load, you need:

- *A list of data quality rules you want to detect errors for. This is usually a subset of the data quality rules, as not all data quality rules are worth spending processing time to detect and handle.*
- *The error-handling specifications. These specify what is to be done when bad quality data is detected. Some errors are so severe that it is worth stopping the load to figure out what is going wrong. Others merely warrant writing the errors to a special holding area where they can be analyzed later.*
- *A development process that links the profiling data quality rules and results to the ETL mapping specifications, and appropriate tool support. Ideally, the detection and handling of data quality is linked to the ETL jobs, but sits externally so that changing the enforced data quality rules does not require rebuilding the entire ETL job stream.*

Summing up: data profiling is crucial to proactive data quality

Many companies are trapped in a cycle that looks like:
Generate poor-quality data → clean it up as best they can → make poor business decisions

To break out of this cycle, a company must identify their core business requirements, the data that is critical to support those requirements, the data quality rules for that data, and see how well those data quality rules match the data. The results of this data profiling can then be used to correct the data (after first determining the cause) and institute processes to keep the data clean from there on out. This proactive approach to data quality can only be achieved through the process of data profiling. Profiling your data is not especially difficult in principle, but it requires a significant commitment from the business and support from IT.

The results of data profiling also bring a significant return on investment in improving the quality of data loads, understanding data lineage and how data was corrupted in the first place (a key regulatory requirement in many industries), streamlining the ETL development process, and enabling Quality Assurance to better detect data anomalies. It can also be a driving force for rigorously capturing metadata and improving the "corporate memory" this way. Further, improved data quality helps to ensure that business initiatives are achieved, compliance is maintained, and that project and business risks are minimized. As with so many things related to Data Quality, it is very difficult to put a dollar value on all these items (especially on the value of improved data quality for decision making), but it is nonetheless there. Just think about this – many analysts at a large bank stated that they regularly spend 25 to 40% of their time trying to figure out how to circumvent poor data quality. You do the math for your company!

David Plotkin is the Data Quality Manager for AAA of Northern California, Nevada and Utah. He is responsible for the implementation of Data Governance, Data Quality, and Metadata. He has 20 years experience in the Data Quality arena, including stints with a major bank and chain drug store. He is a popular speaker at international conferences, and focuses on practical implementation of Data Quality (such as Data Profiling).

For more information

For more information about IBM Information Server, contact your IBM marketing representative or visit **ibm.com**/software/data/integration

IBM Software Group
Route 100
Somers, NY 10589

Printed in the United States
April 2008

TAKE BACK CONTROL WITH **Information Management**