

Data Quality Rules: General Attribute Dependencies

By
Arkady Maydanchik



Editor's Note: This article is the sixth in a series of excerpts from the book *Data Quality Assessment* by Arkady Maydanchik. The book presents systematic, step-by-step instructions for identifying, warehousing, and analyzing data errors. Part 1 of the series can be found in our July 2007 issue. Readers will find a link to the book at our IAIDQ Bibliography page at <http://iaidq.org/main/bibliography.shtml>.

INTRODUCTION

Data represent attributes of real world objects (e.g., people, sales) whose characteristics are interrelated and whose behavior is complex and restricted by numerous logical constraints. These constraints can always be translated into data relationships and used to test data quality.

In the last three articles of this series, we discussed several specific views of the data for complex real world objects. We first investigated the time dimension and the data quality rules arising from the dynamic relationships in the data. We then looked at the lifecycles of state-dependent objects and the rules governing the state-transition data. These investigations, however, still leave unexplored numerous miscellaneous attribute dependencies in the data describing real world objects.

Finding such general attribute dependency rules is more difficult; yet this is key to the success of any data quality assessment project. In this issue, we discuss various strategies and techniques that help us succeed in this challenging task.

TYPES OF ATTRIBUTE DEPENDENCY RULES

We say that two attributes are dependent when the value of the first attribute influences possible values of the second attribute. Consider the relationship between attributes *Hire Date* and *Termination Date* for an employee. Clearly a person must be hired first and terminated later.

This translates into a simple attribute dependency rule:

$$\textit{Termination Date} > \textit{Hire Date}$$

Dependencies between two attributes (sometimes referred to as binary relationships) are the simplest to identify and use. However, often more than two attributes participate in the relationship. In the latter case, the values of the dependent attribute are influenced by the values of the other attributes.

Attribute dependencies generally fall into five broad categories: redundant attributes, derived attributes, partially dependent attributes, attributes with dependent optionality, and correlated attributes.

REDUNDANT ATTRIBUTES

Redundant attributes are data elements that represent the same attribute of a real world object. While attribute redundancy goes against basic data modeling principles, it is common in practice for several reasons. First, redundancy is widespread in “legacy” databases and certain systems that were converted from the “legacy” databases. Secondly, redundancy is often used even in modern relational databases to improve efficiency of data access, information presentation, and transaction processing. Finally, some data across different systems are invariably redundant. Comparison of redundant attributes is a sure way to identify (and eventually correct) numerous data problems.

DERIVED ATTRIBUTES

Values of *derived attributes* are calculated based on the values of some other attributes. This approach is very common when the calculation is rather complex and involves data stored in multiple records of possibly multiple entities. Performing the calculation on the fly is then very inefficient. One of the most common special cases of derived attribute constraints is a *balancing rule*, which requires an aggregate attribute to equal the total of atomic level attribute values.

Derived attribute constraints often find numerous errors. The reasons are largely the same as those for the discrepancies in redundant attributes – it is very hard to create a failsafe yet practical mechanism to keep the data in sync.

PARTIALLY DEPENDENT ATTRIBUTES

The values of redundant and derived attributes are prescribed exactly by the dependency. Oftentimes, the relationships between attributes are not so exact. The value of one attribute may restrict possible values of another attribute to a smaller subset, but not to a single value. We call such attributes *partially dependent*. Any database is full of attribute relationships, and it is simply a matter of patience and analysis to identify many of them. This work always pays off, as the data quality rules for partial attribute dependencies will find large pockets of errors.

CONDITIONAL OR DEPENDENT OPTIONALITY

Conditional optionality represents situations where values of one attribute determine whether or not the other attribute must take Null or not-Null value (i.e., is the value to be prevented or required). Technically speaking, attributes with conditional optionality are a special case of partially dependent attributes discussed above. However, they deserve separate consideration due to their highly frequent occurrence in practice.

A rather trivial special case of conditional optionality is the situation where two attributes are *mutually exclusive* (also called disjoint attributes), i.e. the mere presence of a value of one attribute precludes another attribute from taking a not-Null value and vice versa. An opposite situation is when two attributes can either be both present or absent.

CORRELATED ATTRIBUTES

So far we have discussed situations where values of some attributes somehow restrict allowed values of other attributes. Occasionally the relationships can be more subtle. Values of one attribute can change the likelihood of values of another one, though not firmly restricting any possibilities. We call such attributes *correlated*.

An example from everyday business databases is the correlation between gender and first name. The majority of names are distinctly male or female. Thus there is a definite relationship between these attributes; however, the relationship is not exact in nature. Finding a female named Fred or a male named Rachel is unexpected but not impossible. And in some cases the names are truly common to both genders, as in cases of Terry and Lee. Still the relationship can (and should) be used in a data quality rule. Unlikely pairs of values should be flagged as potentially erroneous.

STRATEGIES FOR DISCOVERY OF ATTRIBUTE DEPENDENCIES

Most of the data quality rules can be identified through rather formal procedures. Attribute domain constraints can be found in data dictionaries and through systematic attribute profiling. Relational integrity constraints are deduced from data models. Timeline constraints for historical data and various rules for event histories can be identified systematically. Rules for state-dependent objects can be inferred from state-transition models.

Still, as the rules get progressively more complex, the search for them becomes more of an art than a science. At the level of event conditions or value patterns for historical data, it takes a creative and inquisitive mind to find the rules. General attribute dependencies fall in the same category. Yet, these more complex business rules are very important as they often find numerous less obvious data errors.

While there is no exact recipe to discover complex attribute dependencies, a few analytical techniques can be used.

1. GATHERING EXPERT KNOWLEDGE

Nobody knows the data better than the users. Unknown to the big bosses, the people in the trenches are measuring data quality every day. And while they rarely can provide a comprehensive picture, each of them has encountered certain data problems and developed standard routines to look for them. Talking to the users never fails to yield otherwise unknown data quality rules with many data errors.

2. INVESTIGATING DATA RELATIONSHIPS

The review and analysis of the data model and other metadata are the keys to finding complex data quality rules. Redundant attributes are usually identified through metadata review. Derived attributes can be identified by reading documentation and analyzing the meaning of each entity and attribute. More complex attribute dependencies can be uncovered with a thorough investigation.

The secret is to systematically consider each data element and inquire:

“If I knew the value of this attribute, what could I say about other data?”

Anytime the value of one data element restricts acceptable values of another data element, we have an opportunity to design a data quality rule.

3. DATA GAZING

Data gazing is simply a process of looking at the data and trying to reconstruct the story in them. Following the real story helps analysts identify parameters about what might or might not have happened. If the story behind certain data values contradicts common sense, you can usually come up with data quality rules to catch the disobedient data.

4. DEPENDENCY PROFILING

Dependency profiling uses computer programs to look for hidden relationships between attribute values. Some of the methods are relatively simple, while others use complex statistical models and pattern recognition techniques. For modern databases, the return on investment from using more sophisticated pattern recognition techniques to identify data quality rules often quickly diminishes. On the other hand, sophisticated dependency profiling is often the best way to identify data quality rules in “legacy” databases with unknown schemas.

SUMMARY

We say that two attributes are dependent when the value of the first attribute influences possible values of the second attribute.

Discovering attribute dependencies, especially those following complex business rules, is more of an art than a science and therefore is quite challenging. However, many analytical techniques can be used, including gathering expert knowledge, in-depth investigation of data relationships, and data gazing.

Formal dependency profiling methods use data mining, statistical models, and pattern recognition techniques to discover hidden data relationships. This work always pays off, as the data quality rules for attribute dependencies will usually find the largest pockets of errors.

About the Author



Arkady Maydanchik is a recognized practitioner, author, and educator in the field of data quality and information integration.

For 10 years, he led Arkidata Corporation, which used his methodology for data quality assessment and data cleansing to provide services to numerous organizations. Arkady is a frequent speaker at The Data Warehousing Institute conferences, IAIDQ conferences, and various data quality seminars.

He has written numerous articles and is author of *Data Quality Assessment*. Arkady also teaches *Practical Skills for Data Quality* courses through his company, Data Quality Group LLC (www.dataqualitygroup.com), and through TDWI (www.tdwi.org/Education/Courses/index.aspx).

He can be reached via email at arkadym@dataqualitygroup.com

© 2007 Arkady Maydanchik