8-16-1996

An Assessment of Database Quality: Design it Right or The Right Design?

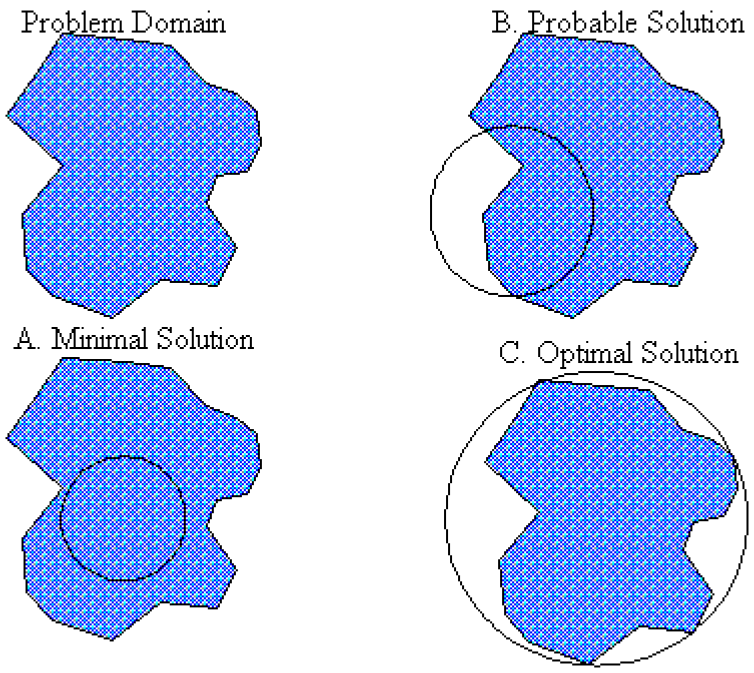
John A. Hoxmeier
Colorado State University

David Monarchi
University of Colorado-Boulder

An Assessment of Database Quality: Design it Right or The Right Design?

John A. Hoxmeier, Colorado State University
David Monarchi, University of Colorado-Boulder

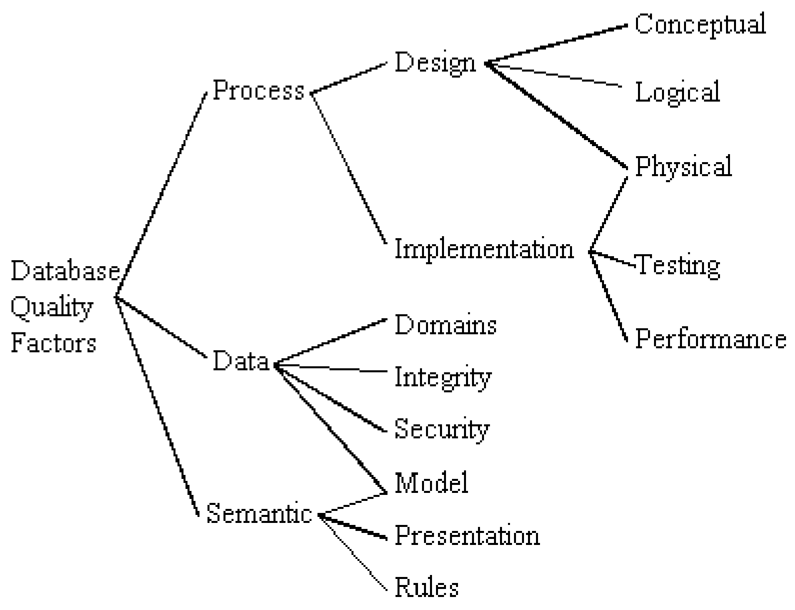
The ultimate objective of database analysis, design, and implementation is to establish an electronic data store that is a model of the relevant aspects of a user's 'world'. Many factors must be considered during this process including, but not limited to, historical and future data perspectives, the diversity of the user community, organizational requirements, security, ownership, performance, temporality, user interface issues, and data integrity. These factors contribute to the success of a database application in either a quantitative or qualitative fashion, and in turn contribute to the overall quality of the database.Figure 1. Database problem and solution domains



What constitutes a database of high quality? Are the criteria different than those used for software applications in general? The process of database implementation consists of the design and construction of a solution domain following the identification of the problem domain (See Figure 1). Because of the difficulties associated with the definition of a fixed set of current requirements and the determination of future utilization, the database problem domain is typically a moving target. As a result, the solution domain rarely approaches the optimal solution presented in Figure 1.c. The database developer must attempt to develop a database model which closely matches the perceptions of the user, and deliver a design that can be implemented, maintained, and modified in a cost-effective way. Software development, in general, is very procedure- or function-driven. The objective is to build a system that works (and do it quickly). Database development, on the other hand, should be more focused on the context and semantics of the data.

To ensure a quality database product, should the emphasis during development be on the application of quality assurance metrics (designing it right)? It's hard to argue against this point, but there is a significant amount of anecdotal evidence that suggests that a quality process does not necessarily lead to a usable database product. A database should be evaluated during implementation based on certain quantitative measures, such as data integrity, normalization, and performance. However, there are also many examples of database applications that are 'well-formed' but lack semantic or cognitive fidelity (the right design). Depending on the application, there may be certain aspects of the quality assessment that deserve heavier weights. Whether the database meets the expectations of its end-users is only one aspect of overall database quality.

There has been a significant amount of literature written on *data* quality. However, there is little available in publications or textbooks on the evaluation of overall database quality. Database quality includes other considerations including process and semantic factors. The balance of these factors is critical in determining overall database quality. Figure 2 presents a possible hierarchy of database quality factors.
Figure 2. Database quality factors



Process Quality

The database design process is largely driven by the requirements and needs of the end-user, who defines the properties of the problem domain and the requirements of the task. The first step is information discovery and it is one of the most difficult, important and labor intensive stages of database development [Chignell, Parsaye, 1993]. It is in this stage where the semantic requirements are identified, prioritized, and visualized. Requirements can rarely be defined in a serial fashion. Generally, there is significant uncertainty over what these requirements are, and they only become clearer after considerable analysis, discussions with users, and experimentation with prototypes. This means previous work may be revisited.

Concentric design is an approach which is appropriate in database design. This cyclical process emulates the philosophy of continuous quality improvement used in Total Quality Management [Braithwaite, 1994]. The costs associated with developing quality into the application from design to implementation are much lower than the costs of correcting problems which occur later due to poor design .

Much attention has been given over the years to process quality improvement. *ISO-9000-3* and *Total Quality Management (TQM)* are approaches which are concerned primarily with the process, not necessarily the outcome [Costin, 1994;

Schmauch, 1994]. *Quality control* is a process of ensuring that the database conforms to predefined standards and guidelines using statistical quality measures [Dyer, 1992]. It compares variations of identified attributes (problem domain) with the results of development (solution domain) and assesses the variation between the two. When deviations from the problem domain are found, they are resolved and the process is modified as needed. This is a reactive form of quality management.

Quality assurance attempts to maintain the quality standards in a proactive way. In addition to using quality control measures, quality assurance goals go further by surveying the customer to determine their level of satisfaction with the product. Conceivably, potential problems can be detected earlier.

The philosophy of ISO-9000-3 is to build quality into a software system on a continuous basis, from conception through implementation. ISO-9000-3 as a process quality standard does not offer any particular metrics to be utilized during the process. In addition, as a general software standard, ISO-9000-3 does not deal specifically with database issues.

All too often, specific performance requirements are either ignored during the design process or evaluated after implementation. While performance, per se, may be more of an implementation issue, it should be considered as an aspect of overall database quality. Both relational and object databases can contain rather serious problems in terms of data redundancy, relationships, integrity, and structure. The objective is to design a normalized, high-fidelity database while minimizing complexity. When evaluating performance there are times when de-normalization of relations may be the optimal solution. The measures used to assess the trade-offs may include query and update performance, storage, and the avoidance of data anomalies. A database that is otherwise well-designed but does not perform well is useless.

Data integrity is one of the keys to developing a quality database. Without accurate data, users will lose confidence in the database or make uninformed decisions. While data integrity can become a problem over time, there are relatively straightforward ways to enforce constraints and domains and to ascertain when problems exist [Moriarty, 1996]. The identification, interpretation, and application of business rules, however, present a more difficult challenge for the developer. Rules and policies must be communicated and translated and much of the meaning and intent can be lost in this process.

Semantic quality is an important objective of database systems. Information that represents a high proportionate match between the problem and solution domains should be the goal of a database with high semantic quality. Qualitative techniques address the ambiguous and subjective dimensions of conceptual database design.

The interaction between people and information is one where human preference and constraints have a huge impact on the effectiveness of database design . The use of techniques such as affinity and pareto diagrams, semantic object models, nominal group, and interrelationship diagraphs help to define the aspects of business requirements which may be elusive. Quantitative techniques, such as entity-relationship diagrams, object models, data flow diagrams, and performance benchmarks, on the other hand, allow the results of the qualitative techniques to be described in a visual format and measured in a meaningful way.

These techniques can be used to assist the developer extract a strong semantic model. However, it is difficult to design a database with high semantic value without significant domain knowledge and experience. In addition, conceptual database design remains more of an art than a science. It takes a fair amount of creativity and vision to design a solution which is robust, usable, and can stand the test of time.

Finally, the assessment of database quality must include value considerations. Time and financial constraints are real concerns. As IT departments are expected to do more with less and as cycle times continue to shorten for database applications, developers must make decisions about the extent to which they are going to implement and evaluate quality considerations. Shorter cycle times present a good argument for modularity and reusability, so quality factors must be addressed on a micro basis.

How does one ensure a final database product that is of high quality? Database quality must be measured in terms of a combination of factors including process quality, data quality, semantic quality, and value. While the balance of these factors will be dependent on the requirements of a specific database application, all should be considered.

References:

Braithwaite, Tim, Information Service Excellence Through TQM, Building Partnerships for Business Process Reengineering and Continuous Improvement, ASQC Quality Press, 1994.

Chignell, Mark and Kamran Parsaye, Intelligent Database Tools and Applications, Wiley, Los Angeles, California, 1993.

Costin, Harry, Total Quality Management, Dryden, United States, 1994.

Dyer, Michael, The Cleanroom Approach to Quality Software Development, Wiley, 1992.

Moriarty, T., "Barriers to Data Quality", Database Programming and Design,, May, 1996, pp. 61.

Schmauch, Charles, ISO-9000 for Software Developers, ASQC Quality Press, 1994.

Teorey,Toby J., Database Modeling and Design, The Fundamental Principles, Morgan Kaufman, San Francisco, California, 1994.

A special thanks to Jane Reyling, CSU, who assisted in the research for this paper.