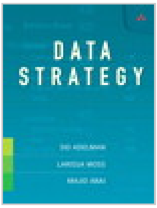
How to Improve Data Quality

By Larissa Terpeluk Moss, Majid Abai, Sid Adelman

Jul 22, 2005

This chapter is from the book



Data Strategy

Learn More Buy

Data Quality Rules

There are four categories of data quality rules. The first category contains rules about business objects or business entities. The second category contains rules about data elements or business attributes. The third category of rules pertains to various types of dependencies between business entities or business attributes, and the fourth category relates to data validity rules.

Business Entity Rules

Business entities are subject to three data quality rules: uniqueness, cardinality, and optionality. These rules have the following properties:

- **Uniqueness**—There are four basic rules to business entity uniqueness:
  - Every instance of a business entity has its own unique identifier. This is equivalent to saying that every record must have a unique primary key.
  - In addition to being unique, the identifier must always be known. This is equivalent to saying that a primary key can never be NULL.
  - Rule number three applies only to composite or concatenated keys. A *composite key* is a unique identifier that consists of more than one business attribute. This is equivalent to saying that a primary key is made up of several columns. The rule states that a unique identifier must be minimal. This means the identifier can consist only of the minimum number of columns it takes to make each value unique—no more, no less.
  - The fourth rule also applies to composite keys only. It declares that one, many, or all business attributes comprising the unique identifier can be a data relationship between two business entities. This is equivalent to saying that a composite primary key can contain one or more foreign keys.

- **Cardinality**—Cardinality refers to the degree of a relationship, that is, the number of times one business entity can be related to another. There are only three types of cardinality possible. The "correct" cardinality in every situation depends completely on the definition of your business entities and the business rules governing those entities. You have three choices for cardinality:
    - One-to-one cardinality means that a business entity can be related to another business entity once and only once in both directions. For example, a man is married to one and only one woman at one time, and in reverse, a woman is married to one and only one man at one time, at least in most parts of the world.
    - One-to-many (or many-to-one) cardinality means that a business entity can be related to another business entity many times, but the second business entity can be related to the first only once. For example, a school is attended by many children, but each child attends one and only one school.
    - Many-to-many cardinality means that a business entity can be related to another business entity many times in both directions. For example, an adult supports many children, and each child is supported by many adults (in the case of a mother and father supporting a son and a daughter).
- **Optionality**—Optionality is a type of cardinality, but instead of specifying the maximum number of times two business entities can be related, it identifies the minimum number of times they can be related. There are only two options: either two business entities must be related at least once (mandatory relationship) or they don't have to be related (optional relationship). Optionality rules are sometimes called reference rules because they are implemented in relational databases as the referential integrity rules: cascade, restrict, and nullify. Optionality has a total of five rules; the first three apply to the degree of the relationship:
    - One-to-one optionality means that two business entities are tightly coupled. If an instance of one entity exists, then it must be related to at least one instance of the second entity. Conversely, if an instance of the second entity exists, it must be related to at least one instance of the first. For example, a store must offer at least one product, and in reverse, if a product exists, it must be offered through at least one store.
    - One-to-zero (or zero-to-one) optionality means that one business entity has a mandatory relationship to another business entity, but the second entity does not require a relationship back to the first. For example, a customer has purchased at least one product (or he wouldn't be a customer on the database), but conversely, a product may exist that has not yet been purchased by any customer.
    - Zero-to-zero optionality indicates a completely optional relationship between two business entities in both directions. For example, the department of motor vehicles issues drivers licenses and car licenses. A recently licensed driver may be related to a recently licensed car and vice versa, but this relationship is not mandatory in either direction.
    - Every instance of an entity that is being referenced by another entity in the relationship must exist. This is equivalent to saying that when a relationship is instantiated through a foreign key, the referenced row with the same primary key must exist in the other table. For example, if a child attends a school and the school number is the foreign key on the CHILD table, then the same school number must exist as the primary key on the SCHOOL table.
    - The reference attribute does not have to be known when an optional relationship is not instantiated. This is equivalent to saying that the foreign key can be NULL on an optional relationship.

Business Attribute Rules

Business attributes are subject to two data quality rules, not counting dependency and validity rules. The two rules are data inheritance and data domains:

- **Data inheritance**—The inheritance rule applies only to supertypes and subtypes. Business entities can be of a generalized type called a supertype, or they can be of a specialized type called a subtype. For example, ACCOUNT is a supertype entity, whereas CHECKING ACCOUNT and SAVINGS ACCOUNT are two subtype entities of ACCOUNT. There are three data inheritance rules:
    - All generalized business attributes of the supertype are inherited by all subtypes. In other words, data elements that apply to all subtypes are stored in the supertype and are automatically applicable to all

    subtypes. For example, the data element Account Open Date applies to all types of accounts. It is therefore an attribute of the supertype ACCOUNT and automatically applies to the subtypes CHECKING ACCOUNT and SAVINGS ACCOUNT.
  - The unique identifier of the supertype is the same unique identifier of its subtypes. This is equivalent to saying that the primary key is the same for the supertype and its subtypes. For example, the account number of a person's checking account is the same account number, regardless of whether it identifies the supertype ACCOUNT or the subtype CHECKING ACCOUNT.
  - All business attributes of a subtype must be unique to that subtype only. For example, the data element Interest Rate is applicable to savings accounts, but not checking accounts, and must therefore reside on the subtype SAVINGS ACCOUNT. If the checking accounts were interest bearing, then a new layer of generalization would have to be introduced to separate interest-bearing from noninterest-bearing accounts.
- **Data domains**—Domains refer to a set of allowable values. For structured data, this can be any of the following:

  - A list of values, such as the 50 U.S. state codes (AL ... WY)
  - A range of values (between 1 and 100)
  - A constraint on values (less than 130)
  - A set of allowable characters (a ... z, 0 ... 9, $, &, =)
  - A pattern, such as a date (CCYY/MM/DD)

  Data domain rules for unstructured data are much more difficult to determine and have to include meta tags to be properly associated with any corresponding structured data. Unstructured data refers to free-form text (such as web pages or e-mails), images (such as videos or photos), sound (such as music or voice messages), and so on. We describe unstructured data in more detail in Chapter 11, "Strategies for Managing Unstructured Data."

Data Dependency Rules

The data dependency rules apply to data relationships between two or more business entities as well as to business attributes. There are seven data dependency rules: three for entity relationships and four for attributes:

- **Entity-relationship dependency**—The three entity-relationship dependency rules are:

  - The existence of a data relationship depends on the state (condition) of another entity that participates in the relationship. For example, orders cannot be placed for a customer whose status is "delinquent."
  - The existence of one data relationship mandates that another data relationship also exists. For example, when an order is placed by a customer, then a salesperson also must be associated with that order.
  - The existence of one data relationship prohibits the existence of another data relationship. For example, an employee who is assigned to a project cannot be enrolled in a training program.
- **Attribute dependency**—The four attribute dependency rules are:

  - The value of one business attribute depends on the state (condition) of the entity in which the attributes exist. For example, when the status of a loan is "funded," the value of Loan Amount must be greater than ZERO and the value of Funding Date must not be NULL. The correct value of one attribute depends on, or is derived from, the values of two or more other attributes. For example, the value of Pay Amount must equal Hours Worked multiplied by Hourly Pay Rate.
  - The allowable value of one attribute is constrained by the value of one or more other attributes in the same business entity or in a different but related business entity. For example, when Loan Type Code is "ARM4" and the Funding Date is prior to 20010101, then the Ceiling Interest Rate cannot exceed the Floor Interest Rate by more than 6 percent.
  - The existence of one attribute value prohibits the existence of another attribute value in the same business entity or in a different but related business entity. For example, when the Monthly Salary Amount is greater than ZERO, then the Commission Rate must be NULL.

Data Validity Rules

Data validity rules govern the quality of data values, also known as data domains. There are six validity rules to consider:

- **Data completeness**—The data completeness rule comes in four flavors:
    - *Entity* completeness requires that all instances exist for all business entities. In other words, all records or rows are present.
    - *Relationship* completeness refers to the condition that referential integrity exists among all referenced business entities.
    - *Attribute* completeness states that all business attributes for each business entity exist. In other words, all columns are present.
    - *Domain* completeness demands that all business attributes contain allowable values and that NULL values can be differentiated from missing values.
- **Data correctness**—This rule requires that all data values for a business attribute must be correct and representative of the attribute's:
    - Definition (the values must reflect the intended meaning of the attribute)
    - Specific individual domains (list of valid values)
    - Applicable business rules
    - Supertype inheritance (if applicable)
    - Identity rule (primary keys)
- **Data accuracy**—This rule states that all data values for a business attribute must be accurate in terms of the attribute's dependency rules and its state in the real world.
- **Data precision**—This rule specifies that all data values for a business attribute must be as precise as required by the attribute's:
    - Business requirements
    - Business rules
    - Intended meaning
    - Intended usage
    - Precision in the real world
- **Data uniqueness**—There are five aspects to the data uniqueness rule:
    - Every business entity instance must be unique, which means no duplicate records or rows.
    - Every business entity must have only one unique identifier, which means no duplicate primary keys.
    - Every business attribute must have only one unique definition, which means there are no homonyms.
    - Every business attribute must have only one unique name, which means there are no synonyms.
    - Every business attribute must have only one unique domain, which means there are no overloaded columns. An overloaded column is a column that is used for more than one purpose. For example, a Customer Type Code has the values A, B, C, D, E, F, where A, B, and C describe a type of customer (for example, a corporation, partnership, or individual), but D, E, and F describe a type of shipping method (for example, USPS, FedEx, or UPS). In this case, the attribute Customer Type Code is overloaded because it is used for two different purposes.
- **Data consistency**—Use the following two rules to enforce data consistency:
    - The data values for a business attribute must be consistent when the attribute is duplicated for performance reasons or when it is stored redundantly for any other reason, such as special timeliness requirements or data distribution issues. Data should never be stored redundantly because of departmental politics, or because you don't trust the data from another user, or because you have some other control issues.
    - The duplicated data values of a business attribute must be based on the same domain (allowable values) and on the same data quality rules.

✚ Share This   Save To Your Account

< Back **Page 3** of 9 Next >