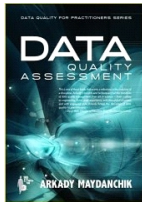


## Data Quality Rules: Relational Integrity Constraints

By  
Arkady Maydanchik



*Editor's Note:* This article is the second 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 appears 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

Of all revolutions in information technology, the introduction of relational data models arguably had the greatest impact. These models gave database designers a recipe for the systematic and efficient organization of data.

Now, some 30+ years since their introduction, relational databases are the cornerstone of the information universe.

Relational data models offer a comprehensive notion of the data structure. In doing so, they also place many constraints on the data. We refer to the data quality rules that are derived from the analysis of relational data models as **relational integrity constraints**. They are relatively easy to identify and implement, which makes a relational data model a starting point in any data quality assessment project. These rules include Identity, Reference, Cardinality, and Inheritance constraints. We discuss each of these rule types below.

### IDENTITY RULES

An **identity rule** validates that every record in a database table corresponds to one and only one real world entity and that no two records reference the same entity.

Imagine a group of pirates dividing the stolen loot according to the personnel table. Mad Dog is engaged in a fight 'til death with recently recruited Mad Doug whose name was accidentally misspelled by the spelling-challenged captain. Wild Billy who changed his name to One-Eyed Billy after the last battle is trying to sneak in and collect two shares in accordance with the register. Life would be tough for pirates in the information age. But it is equally tough for employees, customers, and other entities whose data are maintained in our modern databases.

A reader familiar with database design will naturally ask, "Aren't identity rules always enforced in relational databases through primary keys?" Indeed, according to sound data modeling principles, every entity must have a primary key – a nominated set of attributes that uniquely identifies each entity occurrence. In addition to the uniqueness requirement, primary keys impose not-Null constraints on all nominated attributes.

While primary keys are usually enforced in relational databases, this practice does not guarantee proper entity identity! One of the reasons is that surrogate keys are often created and nominated as primary keys. Surrogate keys use computer generated unique values to identify each record in a table, but their uniqueness is meaningless for data quality.

More importantly, multiple records with distinct values of the primary key may represent the same real world object, if the key values are erroneous. Finding these hidden cases of mistaken identity requires sophisticated de-duplication software. Fortunately, various tools are available on the market for de-duplication of records for persons or businesses.

## REFERENCE RULES

A *reference rule* ensures that every reference made from one entity occurrence to another entity occurrence can be successfully resolved.

Each reference rule is represented in relational data models by a foreign key that ties an attribute or a collection of attributes of one entity with the primary key of another entity. Foreign keys guarantee that navigation of a reference across entities does not result in a “dead end.”

Foreign keys are always present in data models but are often not enforced in actual databases. This is done primarily to accommodate real data that may be erroneous! Solid database design precludes entering such records, but in practice it is often considered a lesser evil to allow an unresolved link in the database than to possibly lose valuable data by not entering it at all.

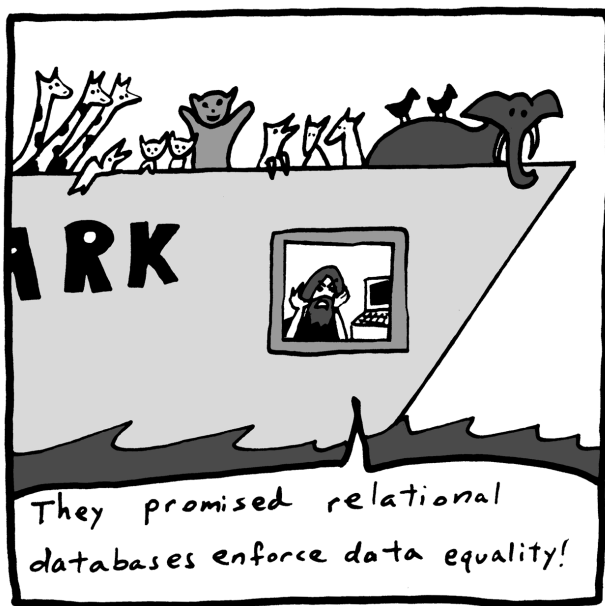
People intend to fix the problem later, but “later” never comes. Foreign key violations are especially typical for data loaded during data conversions from “legacy” non-relational systems, or as a result of incomplete record purging.

## CARDINAL RULES

A *cardinal rule* defines the constraints on relationship cardinality. Cardinal rules are not to be confused with reference rules. Whereas reference rules are concerned with the identity of the occurrences in referenced entities, cardinal rules define the allowed number of such occurrences.

Probably the most famous example of a practical application of cardinal rules is Noah’s ark. Noah had to take into his vessel two animals of each species – male and female. Assuming that he had tracked his progress using a relational database, Noah would need at least two entities – SPECIES and ANIMAL – tied by a relationship with a cardinality of exactly one on the left side and two on the right side.

In fact, Noah’s task was even more complex as he needed to ensure that the two selected species were of different gender – an inheritance rule that we will discuss in the next section. And, of course, he needed to ensure the proper identity of each animal.



I imagine that had Noah used modern technology and had the data quality been consistent with a level that is common in modern databases, we would remember the story of Noah's ark in the same context as the mass extinction of the dinosaurs.

Cardinal rules can be initially identified by analysis of the relationships shown in the relational data models. However relationship cardinality is often represented incorrectly in relational data models.

For example, optionality is sometimes built into the entity-relationship diagrams simply because real data is imperfect. Another problem is that commonly used data modeling notations do not distinguish cardinality beyond *zero*, *one*, and *many*.

Thus, cardinality "many" is used as a proxy for "more than one," without specifying the actual cardinality constraint.

In order to identify true cardinal rules, we use relationship cardinality profiling – an exercise in counting actual occurrences for each relationship in the data model. Once counted, the results are presented in a cardinality chart showing how many of the parent records have 0, 1, 2, and so on corresponding dependent records. A large frequency is usually indicative of legitimate cardinalities, while rare occurrences are suspicious and require further investigation.

## INHERITANCE RULES

An *inheritance rule* expresses integrity constraints on entities that are associated through generalization and specialization, or more technically through sub-typing.

Consider entities EMPLOYEE and APPLICANT representing company employees and job applicants respectively. These entities overlap as some of the applicants are eventually hired and become employees. More importantly, they share many attributes, such as name and date of birth.

In order to minimize redundancy, an additional entity – PERSON – can be created. This new entity houses common, basic, indicative data for all employees and applicants. The original entities now only store attributes unique to employees and applicants. These three entities are said to have a sub-typing relationship.

Inheritance rules enforce validity of the data governed by the sub-typing relationships. For instance, the rule based on the complete conjoint relationship between entities PERSON, EMPLOYEE, and APPLICANT has the form:

*Every person is an employee, an applicant, or both.*

Any occurrence of a PERSON that has no corresponding entry in either the EMPLOYEE or the APPLICANT entities is erroneous (or more likely points to a missing EMPLOYEE or APPLICANT record).

## SUMMARY

Relational data models are a gold mine for data quality rules, specifically Identity, Reference, Cardinality, and Inheritance constraints.

Along with the Attribute Domain constraints discussed in the previous article (*Data Quality Rules: An Introduction* in the July 2007 issue of the IAIDQ Newsletter), Relational Integrity constraints are the easiest to identify and implement.

Unfortunately, such constraints will only locate the most basic and glaring data errors. In the future articles of this series we will graduate to more advanced categories of data quality rules.

## About the Author



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

He is a speaker at the 2007 IDQ Conference in Las Vegas, Nevada.

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](http://www.dataqualitygroup.com)), and through TDWI ([www.tdwi.org/Education/Courses/index.aspx](http://www.tdwi.org/Education/Courses/index.aspx)).

He can be reached via email at [arkadym@dataqualitygroup.com](mailto:arkadym@dataqualitygroup.com)

© 2007 Arkady Maydanchik