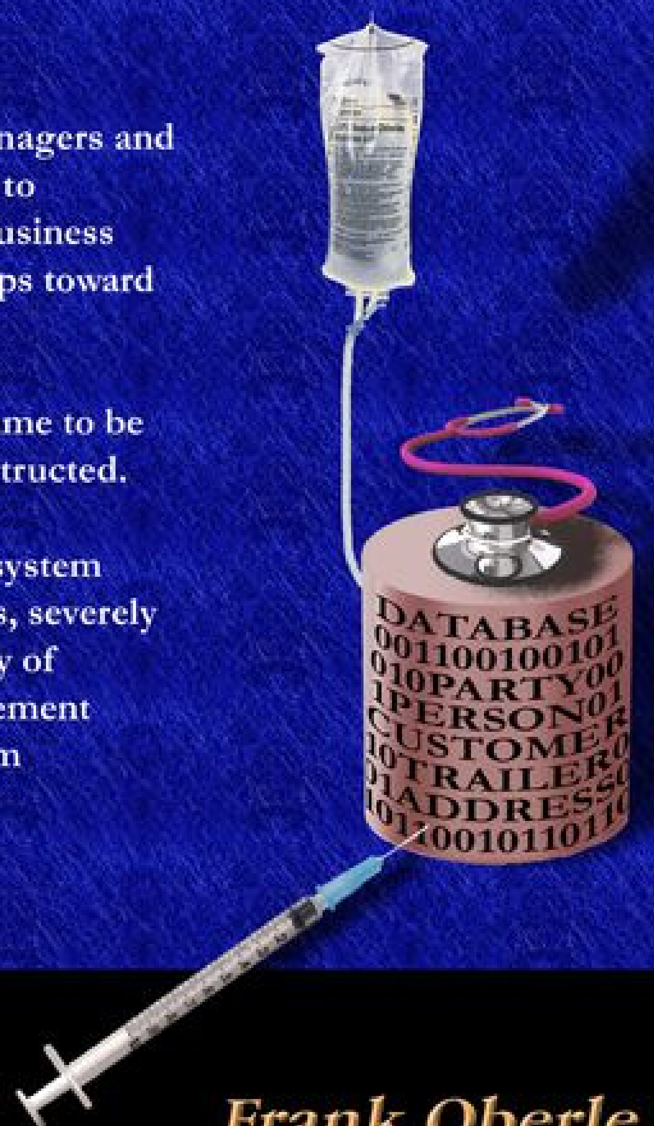
BUSINESS DATABASE TRIAGE

An introduction for both Business Managers and Information Technology practitioners to classifying the symptoms and ills of business databases and how to take the first steps toward treating them.

> Why and how business databases came to be poorly designed and illogically constructed.

> How poor database design inflates system development and maintenance costs, severely limits the flexibility and extensibility of business software, impedes enhancement efforts, and generally leads to System Constipation.



Frank Oberle

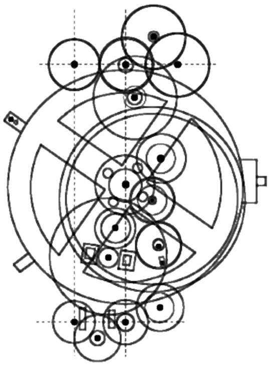
Printed Copies of this Book may be Ordered from the printer, at

https://www.createspace.com/4513537

or from Amazon, at

http://www.amazon.com/Business-Database-Triage-Frank-Oberle/dp/0615916937

www.antikytherapubs.com



BUSINESS DATABASE TRIAGE

An introduction for both Business Managers and Information Technology Practitioners to classifying the symptoms and ills of business databases and how to take the first steps toward treating them.

- Why and how business databases came to be poorly designed and illogically constructed.
- How poor database design inflates system development and maintenance costs, limits the flexibility and extensibility of business software, and generally leads to System Constipation.

Although replete with detailed examples and strategies, this is not primarily a book about database design, nor is it intended to be particularly technical. Rather, this book is an introduction to the fundamental logical principles behind the organization of data, since an unfamiliarity with these principles is one of the primary causes of poor database design.

Frank Oberle

With contributions by Aristotle, Lewis Carroll, Ludwig van Beethoven, Bill Clinton, and other world-renowned data management experts.

Business Database Triage

Publication Information:

Antikythera Publications
e-Mail: antikythera@rcn.com

ISBN-10: 0615916937
ISBN-13: 978-0615916934

Business Database Triage

Aristotle
384 bce – 322 bce

Charles L. Dodgson
27 Jan 1832 – 14 Jan 1898

Bertrand Russell
18 May 1872 – 2 Feb 1970

– Famous but Neglected Data Management Experts of the Past –

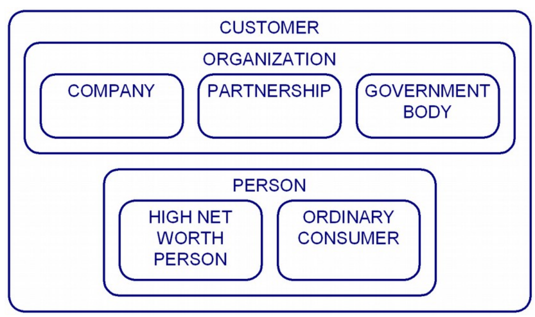
Table of Contents

The "customer-as-organization" versus "customer-as-person" issue illustrated in A Corporate Merger – Part 1 under "Accidents often occur at Intersections" on page 54, for instance, provides one example of the differences between good and bad material, and what to look out for.

In his excellent data modeling book[145], David Hay makes it very explicit that an "order" is essentially a contract between parties, an that a party may be (actually "must be") either a person or an organization of some sort. Although not as explicit, the authors of The Data Model Resource Book[146] likewise address this correctly. These two books (and certainly others) can safely be used as learning material, but many other authors present some solutions that, while possibly workable for single isolated applications (and is there really any justification for any of those anymore?), lead inevitably to poor data architectures not only across the enterprise but also beyond it, and certainly add unnecessary complexity to any applications written to utilize these data structures.

What is a Customer? - Misguided Literature

Presented with an identical "customer-as-organization or customer-as-person" scenario, one author, in what is an otherwise generally useful book[147], and one of the relatively few that are non-product-specific, suggests creating the class structures shown to the right.



Poor Modeling of Subtypes and Supertypes. Don't accept this !

This may or may not represent a convenient *view* of these entities from the standpoint of some particular application, but it clearly conflicts with reality. Consider ***some*** of what this diagram states in logical terms:

145 Hay [1]; see page 383. Chapter 6 of his book (Contracts) begins on page 95.

146 Silverston [1]; see page 384. See Chapter 4 of his book.

147 Simsion [1]; see page 385. See "Subtypes and Supertypes" on page 92.

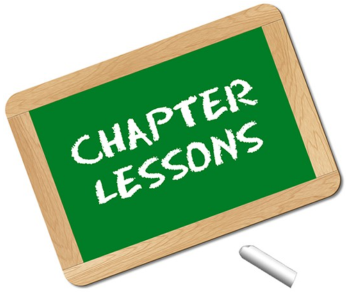
Derived (Normalized) Proposition	Alternate Equivalent Propositions
T All Companies are Organizations	T Every Company **is an** Organization T A Company **must always be an** Organization T Any Company **is/must be an** Organization
F All Organizations are Customers	F Every Organization **is a** Customer F Any Organization **is/must be a** Customer F No Organization **is not a** Customer
Therefore: (based on the above) F All Companies are Customers	F Every Company is a Customer F Any Company must be a Customer
F All Persons are Customers	F Every Person **is a** Customer F Any Person **is/must be a** Customer F No Person **may not be a** Customer
T All Partnerships are Organizations	T Every Partnership **is an** Organization T Any Partnership **is/must be an** Organization
F No Partnership is a Company	F No Partnership **is a** Company F No Partnership **can be a** Company F No Company **is a** Partnership F No Company **can be a** Partnership

In "Mildly Offensive Beliefs" on page 36, I referred to the author's quote "*there is usually more than one way of doing this (classifying data into tables and columns)*" and suggested that the way he selected for his classification of data into tables and columns was logically incorrect.

By reading the author's diagram as if it were a group of Propositions[148], it is easy to see that this analysis is fundamentally flawed and should be summarily rejected as a basis for any further design.

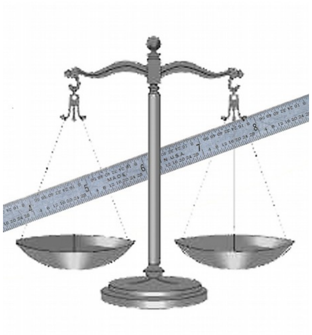
As seductive as it may sound outside of a larger context, Organizations and Persons cannot logically be considered subclasses of Customer. A further difficulty with the model is the author's introduction of "High Net Worth Person" and "Ordinary Consumer" as sub-Classes, which presents the following logical difficulties.

148 As mentioned repeatedly in this book, this is an extremely useful logical quality-control measure – far more useful in most cases than determining the "normal form" of a table.



LESSONS FOR CHAPTER 11
HANDLING OF CONSTRAINTS

- Constraints can be used to enforce Business Rules as well as Data Rules, and if we don't clearly distinguish between the two when designing a database, the oversight will eventually lead to serious problems.
- Constraints on Data Rules are absolutely necessary to protect data integrity, and tend to be factual, specific, and immutable. See "Playing with Trucks – Part 1" to see how poor database design conflicts with this.
- Constraints on Business Rules are likewise necessary, but tend to be arbitrary and subject to change over time.
- Constraints can and often are implemented unintentionally.
- Constraints can be implemented …
    - in a variety of ways
    - in a variety of locations
    - in parallel with other Constraints or in Series with them
- For Data Rules, the "gold standard" for Constraints is that they are
    - implemented in a Declarative manner,
    - implemented in the Database, and
    - implemented in Series with any other Constraints
- From a business perspective, the effectiveness of any Constraints is severely and negatively impacted if there are multiple databases.



Supporting the handling of weights and measures at an enterprise wide level is perhaps the easiest (i.e. least politically sensitive) enhancement to undertake. It is thus recommended as a suitable "first step" towards introducing developers to the many benefits that can accrue from a logical approach to database design.

"Est Modus in Rebus" ("There is Measure in all Things") – Quintus Horatius Flaccus (Horace): Satires i,1

12

12 - Weights & Measures – Part 1

Weights and Measures[221] of various sorts are a key component of most business databases and, indeed, of many types of databases beyond the scope of this book, so it would be remiss not to discuss how these are typically handled and contrast that with how they should be handled. Although at first glance, this manner of properly handling data in accordance with the relational model and predicate logic might seem convoluted, it actually allows both businesses and applications to achieve a much higher level of flexibility, increases data integrity, reduces development time needed to implement future changes, and achieves other objectives dear to the hearts of both IT and business professionals..

Logical approaches for reorganizing other types of data structures to better reflect reality have been presented earlier, but at a rather more general level than many developers might consider useful. As I mentioned earlier, this is not primarily intended to be a database design book, but lack of actual example code can certainly contribute to the idea that the ideas presented here are more theoretical than practical.

221 In the spirit of being precisely "logical," it should be admitted that a Weight is, of course, a specific form of Measure, but given the common acceptance of the term "Weights and Measures" throughout physics and science textbooks, as well as in U.S. and international standards, that term will be used here with only this insincere token apology.

In order to lend credence to the idea that actually implementing logical data structures is not only possible in the "real world," but actually quite straightforward, the subject of Weights and Measures will be dealt with in much greater detail over several chapters, even going so far as to discuss analysis and implementation strategies and to provide pseudo-code[222] examples.

The Business Issue

In "Playing with Trucks – Part 1" (beginning on page 155) as well as in many other scenarios where a business contemplates removing some of the strictures placed on their operations, or even actively joining the global community, implementation of designs based on the approach outlined over this and the next few chapters permits a much higher degree of flexibility and a more rapid and painless response to similar business needs in the future.

Although the scope of this book has been specifically limited to "Business[223] Databases," proper and explicit handling of measurement data is certainly applicable to other fields where data is "explored." Consider the following anecdote, for example:

> In 1999, NASA conducted a research mission in which its Polar Lander was to explore the surface of Mars. Another craft, the Climate Orbiter, would circle the planet and serve as the intermediate navigation and control station, and relay data between the Polar Lander and Earth.
>
> On September 23, 1999, after a 286 day journey, the Climate Orbiter, which cost about $125 million, fired its engine to achieve the desired orbit according to instructions transferred between the Lockheed Martin Corporation in Colorado and NASA's Jet Propulsion Laboratory in California.
>
> Unfortunately, the Orbiter came about 100 kilometers closer to the planet's surface than the engineers intended – and actually about 25 kilometers closer than the altitude at which the Orbiter could even function. As a result, the craft's propulsion system overheated, ultimately causing the Climate Orbiter to be lost.
>
> When looking into the causes for this, it was determined that all data was handled as

222 Well, it's actually very simple (albeit working) PL/SQL code used to test the functionality while writing these chapters, but for non-Oracle users it may as well be pseudo-code.

223 "Business" as defined in Chapter 1, (Definition of "Business"), beginning on page 1.

absolute values – that is, with no stated unit-of-measure. NASA assumed that distances were in kilometers but, unfortunately, the Lockheed Martin engineering team supplied the absolute values with an assumption that the distances were in miles. Oops![224] No Harm, no Foul, apparently. On September 30, 1999, according to CNN[225], the JPL administrator said "No one is pointing fingers at Lockheed Martin."

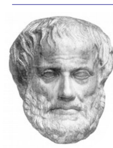
Of course this was just taxpayer money. Most business managers are not so forgiving; the lesson is that removing assumptions can never hurt.

The Objectives of this Exercise

Earlier anecdotal chapters, such as "A Corporate Merger – Part 1" and "Playing with Trucks – Part 1" illustrated some of the difficulties resulting from poor database design. These examples emphasized the necessity for both designing data structures that closely fit the taxonomies of the real-world Things being represented by the data they contain, as well as avoiding hard coding of assumptions into database structures. Both of these practices invariably lead to severe system limitations. Furthermore, hard coding of assumptions into database structures causes these same assumptions to be implicitly hard coded into any applications utilizing them, since there is very little that software developers can do to effectively mitigate this. Unfortunately, the implications of this hard coding is seldom recognized, much less addressed.

Handling of Weights and Measures in typical information systems is one particular area in which hard coding of assumptions occurs quite frequently. This discussion should help expose the extent of these assumptions.

Aristotle Speaks



Remember that the second sentence in his Categories describes "univocal" (unequivocal or unambiguous) naming. The need for this is very close to the root of Philosophy, Logic, and Science, and was recognized as so by Newton, Carroll, etc. Are any of us wise enough to simply ignore this?

Categories; Part 1; Section 1.2

Over the next few chapters, we will discuss what specific data elements are required to

224 ...and this isn't an acronym for Object-Oriented Programming.

225 http://articles.cnn.com/1999-09-30/tech/9909_30_mars.metric.02_1_climate-orbiter-spacecraft-team-metric-system?_s=PM:TECH

unequivocally define the information elements needed for a representative sampling of weights and measures. We will demonstrate that there are numerous steps that can be taken to model this data more effectively in relational databases, and show how to make the transition from existing practices to designs that are more reliable, flexible and extensible, and to do this in an evolutionary manner if necessary. To accomplish these objectives we will, over this and several additional chapters do the following:

In this chapter, we will:

- Outline the design principles and objectives used to guide development of models and processes for storing and manipulating weights and measures.
- Define the various classes of weights and measures that are in and out of scope for this exercise.
- Discuss what data elements are needed to completely and unequivocally define any particular measure.
- Construct a generalized logical model for handling most types of data related to weights and measures.

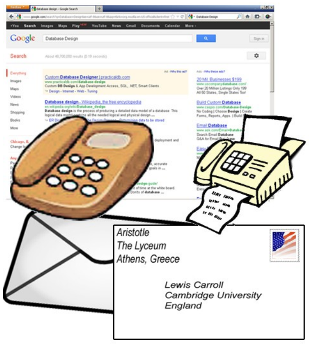
In "Weights & Measures – Part 2" we will discuss the mathematics required for correctly manipulating data relating to weights and measures.

In "Weights & Measures – Part 3" we will describe the desired end state of our system(s) once changes have been made.

In "Weights & Measures – Part 4" we will:

- Outline a process for migrating from typical database and application designs to extensible designs in manageable stages.
- Discuss some of the factors that will need to be specified for any company-specific implementation of these methods.
- Discuss some issues encountered when creating applications that utilize logical data models such as the one presented here.

Finally, in "Chapters 12 & 13 - Weights & Measures Parts 1 & 2" of Appendix B, (page 393) we will:



"Skill is fine, and genius is splendid, but the right contacts are more valuable than either."
Sir Arthur Conan Doyle (1859-1930)

Like the four Weights and Measures chapters, this one crosses the imaginary line into a discussion of design techniques, but only far enough to outline and explain a logical, but non-typical, approach to handling Party-based contact attributes.

18

18 - Contact Mechanisms

Introduction

Throughout many of the previous chapters, the **Party** entity/superclass, although of paramount importance to a well-designed Business Database, hasn't been shown with any attributes but the primary key that is inherited by all of the Party subclasses (e.g. people) participating in the transactions of the business.

This chapter introduces Contact Mechanisms, which are the most common attributes of the **Party** entity in most Business Databases.

As used in this chapter, the term "Contact Mechanisms" refers to any means we have for active or passive communications between Parties in connection with the business they are conducting. If the concept of **Party** isn't clear at this point, it may help to review "An IT Department Must Have Parties," beginning on page 2, once more. Examples of contact mechanisms include telephone numbers of

> Attributes versus Columns
>
> 
>
> There are any number of well-meaning Relational Database design texts stating that any table having nothing but a primary key column can be safely eliminated from the design.
>
> This is nonsense of the first order, possibly originating from the equally silly belief that "Entities become Tables, and Attributes become Columns" in a database.
>
> The reality is that many Attributes become Relationships rather than Columns, and the Party Entity is the most common example of the use of such relationship attributes.

various types, mailing addresses, and so forth. The following list discusses several examples of these.[332]

- **Address**, as used here, is a grouping of data written or printed on any item as directions for delivery to some Party or some Party's specified location. An Address of this type generally falls into one of two broad classes:
  - **Virtual Address**, defined as a description of a location to which certain items may be sent for immediate **or eventual** delivery to a Party (typically a person or organization), whether or not the Party generally resides or can be found at that address. A Post Office Box is an example of a Virtual Address, as are e-mail "addresses," social media "handles" and other such entities.
  - **Physical Address**, a subset of Virtual Address defined as describing a physical location to which deliveries may be made for a particular Party (typically an organization or person), and/or at which the Party may typically be located or reached. A home address is an example of a Physical Address.

  Whether the distinction between Virtual and Physical Addresses is (or may become) important to a particular Business needs to be determined deliberately to avoid introducing any assumptions into the Company's IT systems which might become difficult to compensate for at a later stage of the company's evolution. The distinction isn't often made at the level of data definitions, since most businesses rely on humans to make such judgments as they process orders for shipping for example. Increasing use of "self-serve" ordering, however, particularly from foreign countries, should cause such distinctions to at least be considered when designing or evaluating a system.
- **Device Contact Number**, a grouping of data (usually numeric characters) entered in sequence, and used to establish electronic communications between two or more Parties, their locations, or their electronic devices. The most common example of such an element would be a **Telephone Number**. In normal use, such numbers may possibly need to be combined with

332 At the risk of being repetitious, this is not intended primarily as a design tutorial, but rather to provide enough "straw man" examples and information to expose readers to some of the issues that must be considered when designing database schemas that will be appropriate for logical and extensible support of Business activities.

additional numeric characters to indicate certain exception processing, such as specific routing instructions (e.g. to connect to a "foreign" telephone system), billing information, etc., but those are independent attributes.

- **to a Location**: examples would include:

  … any telephone numbers for an Office or Corporation, a Machine (e.g. fax, modem etc.), a Residence (potentially associated with multiple persons).

  … telephone numbers for Alarm systems (1-way, not 2-way), whether land-line or cellular.

  … telephone numbers used to connect to other devices (1-way as well as 2-way), whether land-line or cellular. Examples would be wireless Hot Spots (fixed or location independent), household cellular devices linking various wired handset instruments in a household to the telephone infrastructure, etc.

  … IP Addresses or MAC Addresses used to route any data communications to a specific device, particularly where the device is associated with some Party.

- **to a Person**: examples include telephone numbers for Cellular Phones that are Party-specific, but location independent (typically for an individual person).

► **Broadcast**

- **One Way**

  Television and Radio channels (always by a Party – usually by a Company).

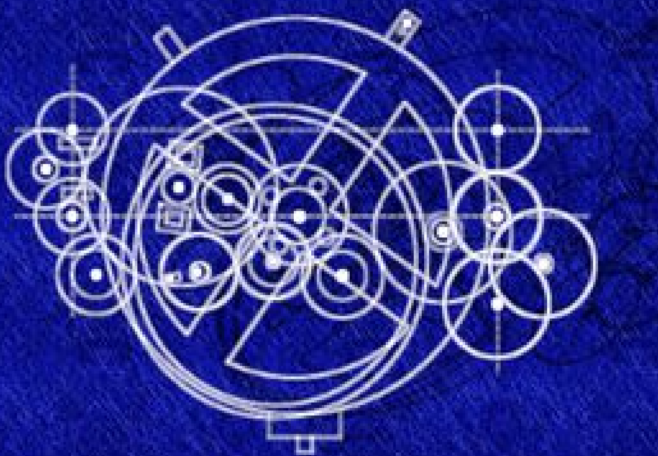
  URLs (Web pages, Podcasts, Blogs & such; done by any subclass of Party).

  Media advertising of various other types, e.g. billboards, and other signage, which occur in far too many forms and locations to consider listing.

- **Two Way**

  Interactive Web Pages, e.g. customer service sites, and the like

The preceding is, of course, not a complete listing of contact mechanisms, nor is it likely to perfectly match the needs of any specific business, and it is the analysis team's responsibility to identify as many potential contact mechanisms as possible while evaluating a database design for some particular business. And, of course, the purpose of such contact mechanisms as well as the technologies used for these mechanisms need to be considered in light of a particular company's needs. Further, the above list doesn't consider the content of messages to



Triage and First Aid for Business Systems - inspired by the teachings of Aristotle as well as those of Beethoven, Carroll, Boole, Russell, Clinton and other philosophers and data organization experts throughout history.

Though seemingly replete with detailed examples and strategies, this is not intended as a book about database design per se. Nor, aside from a few examples where some familiarity with SQL may be helpful, is this book intended to be particularly technical.

Rather, this book is an introduction to the fundamental logical principles behind the organization of data - a critical responsibility of both IT Technologists and Business Managers. Experience suggests that many of these principles - which form the foundation of all the Sciences - are as unfamiliar to IT Personnel (including many of those tasked with designing business databases) as they are to Business Managers.

> This book is therefore aimed at both those groups.

90000

9 780615 916934

Antikythera Publications

Databases
Information Management
Business

www.antikytherapubs.com