JOE CELKO'S DATA, MEASUREMENTS AND STANDARDS IN SQL

MK
...N KAUFMANN

Joe Celko's Data, Measurements and Standards in SQL

Joe Celko

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Morgan Kaufmann Publishers is an imprint of Elsevier

MORGAN KAUFMANN

Morgan Kaufmann Publishers is an imprint of Elsevier.
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
This book is printed on acid-free paper.

Library of Congress Cataloging-in-Publication Data

Celko, Joe.

Joe Celko's data, measurements, and standards in SQL/Joe Celko.

p. cm.—(The Morgan Kaufmann series in data management systems)

ISBN 978-0-12-374722-8

1. SQL (Computer program language) 2. Database management. I. Title. II. Title: Data, measurements, and standards in SQL.

QA76.73.S67C4324 2010

005.74—dc22

2009030587

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

For information on all Morgan Kaufmann publications, visit our Web site at www.mkp.com or www.elsevierdirect.com

Printed in the United States of America

09 10 11 12 13 14 15 16 5 4 3 2 1

Introduction

As I sit here typing this introduction, I am looking at a copy of *The North American Arithmetic, Part Third, for Advanced Scholars,* by Frederick Emerson. It was written in 1834.

It has chapters on the money, weights, and measures in use in Europe and America at that time. The English still had pence, shillings, and farthings. The countries and city-states on the continent had a local currency instead of a Euro. Troy, Avoirdupois, and Apothecary weights were mentioned as standards in Europe.

But after that, there were English ells, French ells, and Flemish ells for measuring cloth—all of them different. There were separate liquid measures for wine and beer. Cities had local measures, so "100 lbs" in Trieste was 123.6 pounds Avoirdupois and 74.77 pounds Avoirdupois in Rome. The good news was that the new metric system in France seemed to be catching on and becoming more popular. How, do you suppose?

All of these differences were handled by Gaugers in each city. These were usually government tax employees who kept a set of dip sticks (or gauges) to measure liquids in barrels, weights for balance scales, baskets for dry goods, and so forth. The term survives today in the oil refinery business for people who test oil in storage tanks. They also calibrated the local merchant's tools to assure fair trade within their jurisdiction. The seal cast on the front of a wine carafe is a leftover of a Gauger's official seal, which would have been stamped into a piece of lead.

Remember that all of this was a long time before computers and electronic data exchange. Think about the manual records needed. You can imagine what effect this had on commerce, since it made accurate data almost impossible to keep and exchange.

The bad news is that this "local measurement" mentality has not died in the 21st century. Ignorance of international and industry standards and a "cowboy coder" mind-set led programmers to code first and think never. Instead of looking for an ISO or industry-specific standard for something, the "cowboy coder" gallops in and writes a list or uses a proprietary auto-numbering feature in his current SQL product to give him an encoding for an attribute.

Programmers still spend time and effort avoiding the use of common units of measure and industry standards. It can actually become quite silly.

In a posting on the Microsoft SQL Server Programming newsgroup, an MVP ("Most Valuable Programmer"—an award given by Microsoft in their product categories) posted a complaint that passing temporal data that will work across regional settings doesn't work in the ANSI Standard! His example was two values on a machine configured to Microsoft's U.K. formats instead of to ISO standards, thus:

```
CAST('2007-04-01' AS DATETIME) - ISO/ANSI SQL format
versus
CAST('2007-04-01T00:00:00' AS DATETIME) - set to UK format
```

To quote: "Here in the UK in cloudy Harpenden the default connection settings give these results ...":

```
'2007-01-04 00:00:00.000'
versus
'2007-04-01 00:00:00.000'
```

The use of a T is an alternative ANSI standard, which makes a timestamp into a single unbroken string (this can be useful in some systems). But the "T" is not part of SQL's subset of the ISO-8601 standard (this might be changed by the time you read this book). But the ordering of year-month-day is in all ISO-8601 standards—and it makes sense for sorting!

I have run into this problem with a system that collected data from the United States and the United Kingdom. You would be surprised how long you can screw up the database before you catch the error.

The poster ranted (imagine strong British accent): "At what point will this guy [Celko] stop telling us duff information and bashing people for not following best practice and standards when he himself knows there's a problem yet refuses to yield to the correct best practice?"

It was very hard to tell him that Harpenden, UK, is not the center of the world, whether it is cloudy or not! Even Greenwich Time is no longer the standard. The best practices are to store time in UTC and use the TIMEZONE options in SQL to get local lawful time in the applications. It is all there in the standards.

COMMON UNITS EVOLVE TO STANDARDS

There is a common pattern in moving to standards, which is easy to see in physical goods. Since beer is one of my favorite physical goods, consider this history.

Units of measurement become standardized and replace local measurements. But we do not have standard sizes yet. For example, after a few hundred years, we get rid of pints, firkins, and a host of local units with strange names to use liters for our beer. But every pub still has a different size glass.

Enterprises working in the same area agree on standard sizes for common use. For example, the pubs all get together in one town and agree that a glass of beer is 0.50 liters; in other town, they decide that a "fair pour" as it is called in the bar trade, should be 0.75 liters (nice place to visit).

National standards emerge, usually by law based on the need for taxation. Continuing my example, the government decided to pass a beer tax based on the glass, so we need a National Fair Pour Standard.

International standards emerge, usually by agreement based on the need for trade and data exchange.

Notice that the glassmaker has a strong influence with his offerings to the bar trade. One trade affects another. Let me give a specific example. Before the 1800s, machine screws were made by individual machine shops in the United States. The common units of measure were inches for the physical dimensions such as length, pitch, and diameter and degrees for the angle of the thread. But that did not mean that the screws were interchangeable or even used the same names for various sizes. It was very much a cottage industry.

Some national standards did eventually arise within industries inside the United States, but it was not until World War II that American, Canadian, and British manufacturers decided to standardize their inch-based sizes for the war effort. The Unified Thread Standard was adopted by the Screw Thread Standardization Committees of Canada, the United Kingdom, and the United States on November 18, 1949, in Washington, DC, with the hope that they would be adopted universally.

It did not work out that way. Europe and much of the rest of the world turned to the ISO metric screw thread system. Today, globalization has forced the United Kingdom and the United States to use this same system. The best example is that the ISO metric screw thread is now the standard for the automotive industry and that very few parts use any inch-based sizes, regardless of where they are made. The United States used to completely dominate the auto industry but lost out for various reasons. A major factor is that ISO machinery is engineered to the millimeter while the U.S. auto industry was working at 1/32 of an inch. An "ISO automobile" is simply more precise and accurate than an "Imperial automobile" because of the measurements used. This was not a problem with the Model T, but it is a disaster with a modern vehicle.

WHAT DOES THIS HAVE TO DO WITH DATABASES?

Today, the database designer is finding himself in the position of the Gauger from the Middle Ages and Renaissance, only his role is greatly expanded.

We have to share and move data, rather than physical goods, from one enterprise to another accurately. The enterprise is not in the next village but anywhere on Earth. We all have to "speak a common language" with the rest of the world.

We still deal with a lot of physical measurements, but we don't use dip sticks to convert "London firkins" to "Harpenden fuddles" anymore. We need a bit more math and have rules about accuracy and computations.

The good news is that we have the Internet, ISO, the EU, ANSI, DIN, JIS, and scores of other organizations that do nothing but help us with standards. These days, many of those standards come with a description of how they are represented in a database. They come with Web sites so that we can download and update our systems for little or nothing. They are easy to find—God bless Google!

And if we don't like them, these same organizations have mechanisms for submitting comments and proposals. It the old days, you had to appeal to the king to become the royal barrel maker, bottle blower, or whatever.

The purpose of this book is to turn a new database designer in a 21st-century Gauger. He is going to need some new skills, in particular:

Knowledge of standards
How to migrate as standards evolve
How to create and use a data dictionary
How to maintain data integrity
How to maintain data quality

In *MAD* magazine #33, Donald E. Knuth wrote a parody of the metric system while in high school, long before he became one of the greatest computer scientists of all time. The title was "Potrzebie System of Weights and Measures," and it was illustrated by Wally Wood (see next page for the image).

Potrzebie is a Polish noun ("potrzeba" translates as "a need") that *MAD* magazine used as a nonsense word in their articles and cartoons.

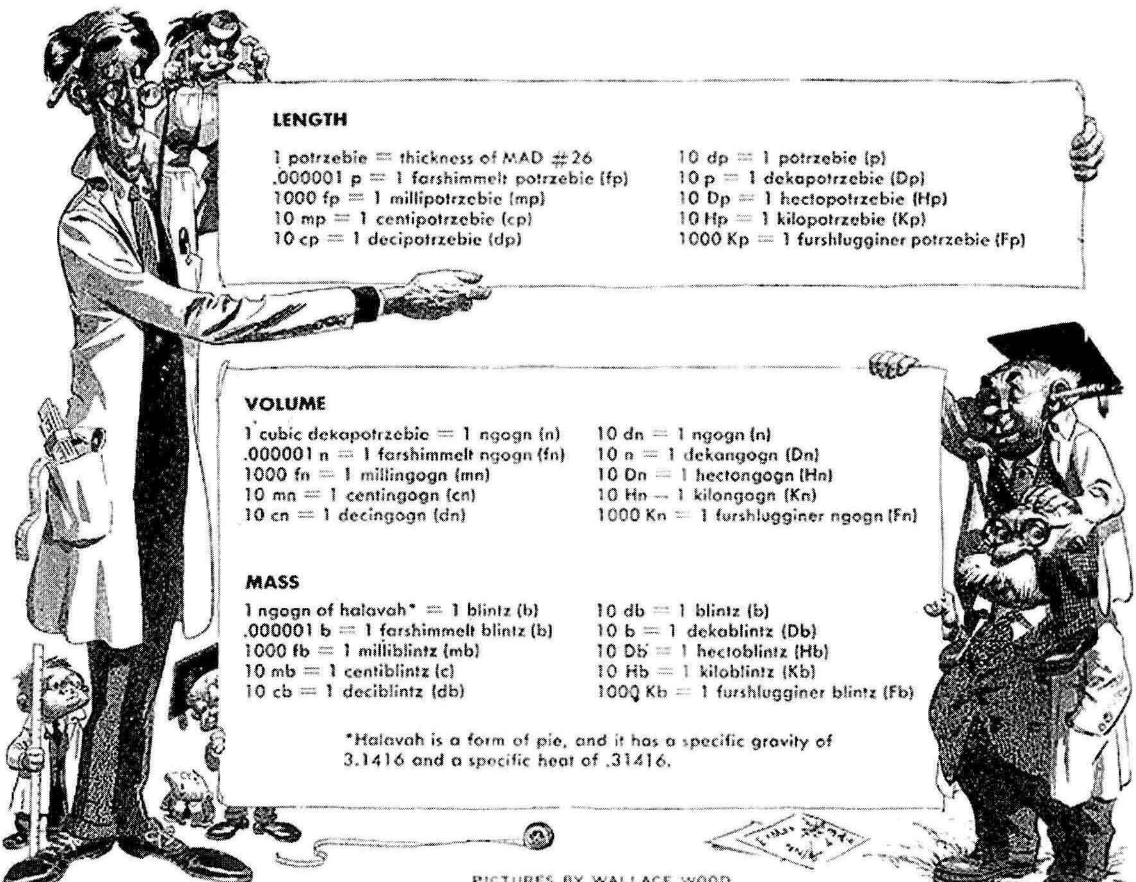
While Dr. Knuth was being funny, the number of database designers who invent elaborate measurement and encoding schemes when standards exist is not so funny.

THE POTRZEBIE SYSTEM
OF WEIGHTS AND MEASURES

THE POTRZEBIE SYSTEM

This new system of measuring, which is destined to become the measuring system of the future, has decided improvements over the other systems now in use. It is based upon measurements taken 6-9-12 at the Physics Lab. of Milwaukee Lutheran High School, in Milwaukee, Wis., when the thickness of MAD Magazine #26 was determined to be 2.263348517438173216473 mm. This length is the basis for the entire system, and is called one potrzebie of length.

The Potrzebie has also been standardized at 3515.3502 wave lengths of the red line in the spectrum of cadmium. A partial table of the Potrzebie System, the measuring system of the future, is given below.



LENGTH

1 potrzebie = thickness of MAD #26	10 dp = 1 potrzebie (p)
.000001 p = 1 farshimmelt potrzebie (fp)	10 p = 1 dekapotrzebie (Dp)
1000 fp = 1 millipotrzebie (mp)	10 Dp = 1 hectopotrzebie (Hp)
10 mp = 1 centipotrzebie (cp)	10 Hp = 1 kilopotrzebie (Kp)
10 cp = 1 decipotrzebie (dp)	1000 Kp = 1 furshlugginer potrzebie (Fp)

VOLUME

1 cubic dekapotrzebie = 1 ngogn (n)	10 dn = 1 ngogn (n)
.000001 n = 1 farshimmelt ngogn (fn)	10 n = 1 dekangogn (Dn)
1000 fn = 1 millingogn (mn)	10 Dn = 1 hectongogn (Hn)
10 mn = 1 centingogn (cn)	10 Hn = 1 kilongogn (Kn)
10 cn = 1 decingogn (dn)	1000 Kn = 1 furshlugginer ngogn (Fn)

MASS

1 ngogn of halavah* = 1 blintz (b)	10 db = 1 blintz (b)
.000001 b = 1 farshimmelt blintz (b)	10 b = 1 dekablintz (Db)
1000 fb = 1 milliblintz (mb)	10 Db = 1 hectoblintz (Hb)
10 mb = 1 centiblintz (c)	10 Hb = 1 kiloblintz (Kb)
10 cb = 1 deciblintz (db)	1000 Kb = 1 furshlugginer blintz (Fb)

*Halavah is a form of pie, and it has a specific gravity of 3.1416 and a specific heat of .31416.

PICTURES BY WALLACE WOOD

36

Let's get started. The first part of this book is the preliminaries and basics of measurements, data, and standards. The second part is the particulars for standards that should be useful for a working SQL programmer.

Please send any corrections, additions, suggestions, improvements, or alternative solutions to me or to the publisher.

Morgan Kaufmann Publishers
30 Corporate Dr #400
Burlington, MA 01803

Contents

Part 1

History, Standards, and Designing Data

The first of this book discusses the principles of designing data encoding schemes and some of the history of standardization. This second part will give examples of actual standards used in a variety of industries.

In the first six chapters, I look at the foundations from the view point of a database designer who needs some understanding of the how and why.

I find it odd that database designers are very physical about their data and do not work with many abstractions. They were never taught the theory of scales and measurements. They have only a minimal knowledge of validation, verification, and risk of error as a part of the data.

Check digit algorithms are taught as single "programming tricks" in undergraduate computer science classes rather than a mathematical discipline.

To the best of my knowledge, I am the only person who teaches Data Encoding Schemes in an orderly fashion.

The failure of cowboy coders to use standards leads to problems. The homegrown encoding schemes have to be