We continue our exploration of database design in part four of this 14-part series, Process-Driven Data Design.

KEYFIELD DESIGN

by Frank Sweet

Database design consists of modeling the business world. Each vendor record should reflect the characteristics of real vendors, and there's one data element that every vendor record has: a keyfield or vendor ID number. After all, that's the way you tell them apart. A moment's thought reveals that keyfields should be dataless, unchanging, unambiguous, and unique. A handy way of remembering this is $K = DU^3$.

Keyfields should be dataless. "Vendor number?" asks our user. "Well, it should contain region code, industry type, size code, purchasing-agent code...." Why do some users insist on embedding clusters of data into ID numbers? Few design issues degenerate so swiftly into ultimata and recriminations between designers and users. There are explanations as well as solutions.

But first, examine the phenomenon itself. It leads to two major problems: the first is nature's 90-10 rule, the second is that data do change.

Nature's 90-10 rule says that 90% of whatever occurrences you're measuring are produced by 10% of the population. If you insert classification schemes into keyfields, you'll run afoul of the 90-10 rule. For example, consider the Case of the Cargo Tracking System:

The application kept track of ocean freighters carrying food products around the world. Each record represented a shipment with such data items as ship name, cargo quantity (in tons), and value (in dollars). The record's four-part keyfield was designed as "CF126523," where the first two bytes indicate the type of cargo from a table of a few dozen types. ("CF," by the way, means "coffee beans.") The next two bytes show the port of origin from a table of seaports, and "12" means Santos, Brazil.

The third slice of keyfield shows the shipment's destination port, from the same table of seaports; "65" means Jacksonville, Fla. Finally, there's a two-byte sequence number. In the example, it means that shipment "CF126523" is the twenty-third shipment on file carrying coffee beans from Santos to Jacksonville.

The 90-10 rule simply says that most coffee is shipped from Brazil (or somewhere in South America) to Jacksonville (or somewhere where there's a roasting plant). Packaged foods, on the other hand, leave processing plants outbound for distribution sites.

This yields two consequences. First, many possible numbers will never be used. Just as you wouldn't bring lobsters to Maine, no one would send coffee beans to Brazil. Second, and more important, is that in the normal course of business, hundreds of shipments carry them from Santos to Jacksonville.

In other words, though there were unused gaps in the numbering scheme, the heavily used sequences (e.g., "CF1265...") soon exhausted all possible numbers. The system had not been up two years before there were 99 shipment records, most of only historical interest, from Santos to Jacksonville. How did we code the hundredth one?

There is no clean solution. The two answers usually proposed for this problem are to make the sequence-number portion of the keyfield longer or to add duplicate entries to one of the tables. The former idea soon foundered on the grim reality of changing the length of a master file's keyfield. It simply could not be done without retrofitting the entire application and its associated administrative procedures. In short, it would have taken too long. If the system were to survive, that hundredth shipment had to be recorded immediately.

We adopted the latter idea. We added code "CO" to the cargo-type table with the same meaning as "CF." This postponed the day of reckoning for another two years. It also transformed the system into a maintenance nightmare. Think of all those summaries by cargo type that could no longer be produced by sorting the records.

The 90-10 rule problem has nothing to do with the food industry or even with event tracking. It is most often found in part-numbering schemes for spare parts in heavy industries like steel or chemicals, or for components in manufacturing. It's incredible how many different kinds of nuts and bolts there are.

2 CHOICES, BOTH ARE WRONG

We see the second pitfall, the unchanging keyfield, by considering what happens when a shipment is diverted. Shipment CR126523 is on its way when dispatching decides its Florida-bound cargo is more urgently needed in Baltimore. But destination-port is part of the record's keyfield. When a shipment is redirected we're faced with two choices—both wrong.

If we leave the shipment's ID number unchanged, its destination-port code becomes inaccurate. Not that it is redundant, inconsistent, untimely, or anything less. It is simply flat-out wrong. Anyone who uses this information will be misled. Worse, it's one of the most important pieces of data in the record. Why else did our user want it in the keyfield in the first place?

If we change the ID number we lose the shipment's audit history. Understand, most data are not in computer files. Shipment numbers are in letters, telephone note pads, contracts, scrawled on the backs of envelopes, on countless 3-by-5 file cards, and in the skulls of users everywhere. If we