subjects:
- News
- Daily News
- Insights
- Reviews
- Special Reports
- eCRM Products
- eCRM Glossary
- IT Management Blog
- Events
- Advertising Info
- Technology Jobs
- Staff

internet.commerce

Be a Commerce Partner

CRM, Data Quality and the Cloud

By David Taber
March 28, 2011

Share 0 Print Email

With most data, you expect a unique key and an unambiguous match. But what do you do when you don't have a key to match on?

In the good old days, every table had a unique key and table joins were a literal match of keys. Even across systems, you could get an unequivocal match that meant valid data.

Related Articles

- Data Virtualization Links CRM, ERP and SCM
- The 10 Most Important CRM Reports
- Fortune 1000 Firms Could Increase Revenue by Billions with Effective Data: Study
- Oracle Aims for Real-Time Business Intelligence
- Appirio's CloudWorks Aims to Break SaaS Logjam

But cloud computing by definition means more loosely-coupled systems. And if you're working with public clouds, there may not be usable external keys. You have to depend on matching data on less-certain criteria, often resorting to fuzzy matches that depend on data context and semantics. This requires some new tools, some subtlety, and craftiness.

Fortunately, customer relationship management (CRM) data lends itself to this kind of matching ... if you're careful and have reasonable expectations. The trick is knowing how to leverage several layers of criteria, and setting the thresholds to maximize matches without an undue number of false positives. So get comfortable with probability and statistics, as opposed to certainty.

The data matching problem

The problem is that there aren't universal taxonomies, names, keys or semantics for most objects you'd want to match across clouds. Sure, there are DUNS numbers, FEINs, SSNs, and other unique identifiers for people and companies, but some of the most useful ones are off limits because of privacy laws like HIPAA, FERPA, and directives from the European Commission. Even when you can use those unique identifiers, their quality degrades with age (due to mergers, spinouts, name changes, etc.).

Technology has made this "identifier problem" worse, not better. VoIP and number portability mean that telephone area codes / city prefixes are no longer reliable indicators of location. Email addresses suffer from "information rot" of as much as 10% a month. Even Salesforce.com's Jigsaw, arguably the biggest publicly available database of email addresses, only covers corporate emails. Since many of the customers you care about the most have multiple email addresses (executives may have 10 or more), it can be very tough to correlate the different email identities. When it comes to consumer email domains (e.g., gmail, yahoo, gmx, freemail, etc.) essentially nobody outside of the NSA, CIA, or FBI can use those as good identifiers. And by definition, one-use emails are useless for matching.

Fuzzy logic and other solutions

If you're lucky and have a lot of fields to match a pair of records, you can use a fingerprinting strategy. For example, you may never know who's at the keyboard, but you could interrogate a browser's URL history to create a fingerprint for each web site visitor. For example, a visitor today may be a close match to someone who registered on one of your sites a month ago. Fingerprinting strategies almost always involve some sort of scoring system, and their effectiveness depends upon the semantic intelligence of the scoring criteria. Even when these systems work, they require care and feeding to keep the scoring coefficients effective.

In most cases, you don't have enough data for the fingerprinting strategy. This is where fuzzy logic comes in: using close matches of a few fields to make the join. You really want to understand the elegance (or not) of the particular fuzzy algorithm before you trust it. At one extreme is the SQL LIKE command, which is simple, free, fast, and yields poor results. At the other extreme are complex AI algorithms that are hard to understand, probably require tuning, and can be as easily misapplied as CDO and CDS valuation algorithms of recent Wall Street infamy.

When using fuzzy match algorithms, apply them in layers, measuring match-quality results at every later to optimize the final result:

- First, clean the data of obvious garbage entries and wild points. You might lose 1% to 10% of your samples, but this will make the rest of your tuning easier.
- If part of your match includes text (such as city, state, country), separate the records by language. Then apply spelling correction only to text fields using the tightest possible criteria. Do not apply spelling correction on peoples' names.
- Now use your fuzzy matching algorithm with the tightest possible criteria. This might be a tool or a piece of your own code, but at this pass you want to accept only the smallest variances (e.g., ignore spaces and punctuation marks).
- Next layer, use fuzzy matching with name substitution (e.g., Jim vs. James, Jack vs. John, Bill vs. William). As before, be careful about language differences — it's best to process each language with a separate dictionary and tuned criteria.
- Next, use a name-substitution scheme for company names (e.g., IBM vs. I.B.M. vs. International Business Machines). This almost certainly will be a table-driven lookup, and you might have to code this yourself.
- If you are trying to match product names, use a similar lookup scheme to allow matches of "slang names" (e.g., "premium BMW sound system" or "spring promotion" vs. "Blaupunkt AVF-30").
- If you're still not getting a high enough match percentage, do some experimentation with loosening only one of the fuzzy match criteria at a time.

At each layer, someone needs to analyze the results for the number and quality of matches. In most CRM use-cases, a 2% "bad match" ratio is perfectly acceptable, and in some cases 5% is still usable.

Which tools to use? Depends on how much you like coding, and how much time you have to spend. Any time you're doing fancy fuzzy stuff on a large data set, a scripting language will be dreadfully slow, even if it's compiled. Off-the-shelf tools may be more limiting than what you build yourself, but they'll be more reliable and give more predictable/consistent results. Perhaps more important, they will have a user community that you can tap for tips and consulting talent to improve the quality of your matches.

David Taber is the author of the new Prentice Hall book "Salesforce.com Secrets of Success" and is the CEO of SalesLogistix, a certified Salesforce.com consultancy focused on business process improvement through use of CRM systems. SalesLogistix has more than 50 clients in North America, Europe, Israel and India. David has more than 25 years experience in high tech, including 10 years at the VP level or higher.

For more by David Taber, see The Truth About Sales Leads and CRM and the Named Account Model - Square Peg, Round Hole?

Share: Digg Del.icio.us Reddit Facebook Twitter

0 Comments **(click to add your comment)**

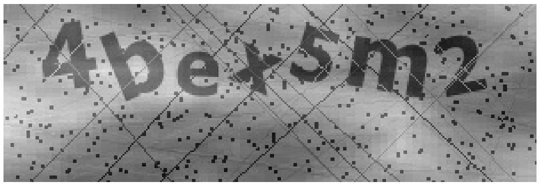
Comment and Contribute

Your name/nickname

Your email

XHTML: You can use these tags: <b> <u> <i>

(Maximum characters: 1200). You have 1200 characters left.



Please type the alphanumeric characters above and click "Submit" to continue.What's this?

I cannot read this. Please generate a New Image

See our comment policy.

Submit Your Comment

The Network for Technology Professionals

Search: Find

About Internet.com

Legal Notices, Licensing, Permissions, Privacy Policy.

Advertise | Newsletters | E-mail Offers