CIO

Print Article Close Window

From: www.cio.com

CRM Records: Dummy Tricks for Smarter Searches

– David Taber, CIO

September 28, 2010

Earlier this month, I wrote about using a few Dummy records in tables to make CRM systems smarter. This time, we're looking at a different situation: Dummy fields that may be needed in all of a table's records.

In most CRM systems, the Search function is pretty limited. Almost all systems have the ability to do simple Boolean keyword searches, but the results are often too narrow and exclude "obvious" items that the users need. Further, most CRM systems' report engines aren't very smart about having intuitive Booleans for filters that can work on any field in a table.

Putting those "invisible" items into a SearchDummy field can make your reports smarter and searches less frustrating. If you'll again forgive me some more Jeff Foxworthy shtick here, let's walk through some examples:

If You Can't See the Name of a Referenced Object, You Might Need a SearchDummy. Even though the CRM screens and reports present related items as human-readable strings, they are typically stored as a pointer referencing another table (e.g., the "account" field in the contact table is actually a foreign key from the account table). So if users try to search for all the people who work at an account, the result set will be meager or even empty. They don't want to (or may not be able to) write a report just to quickly spot who works at XYZcorp, particularly when they don't remember exactly how to spell the company name.

A SearchDummy field can automatically include the string "XYZcorp" in every relevant contact, opportunity, case, or any other related table that references accounts, so those related records will show up in a search.

If you Can't Search for Pick List Values, You Might Need a SearchDummy. Pick list values are always presented as human-readable strings, but your CRM system may not store them that way. Consequently, when users try to search for "XYZcorp" AND "manufacturing" (the industry/sector pick list) the result set will always be empty. While in theory the users could write a report, why not make the system handle these ad-hoc searches all the time?

If You Can't Search for Numerical Ranges, You Might Need a SearchDummy. While your CRM system surely stores numbers as numerical values, there's no way to tell search for all companies with sales between $100 and $250 million. Some search systems won't let you search for numbers at all (as this can generate some seriously silly results).

You can set up a SearchDummy to flag numerical ranges on a few variables that your users need to screen for. The SearchDummy field will contain strings that users will intuitively use as keywords, such as "$100-250 M revenues" or "Last 90 days." Sure, you'll need to create a cheat-sheet of keywords...but this can be a real time-saver.

If You Can't search for Exceptions or Workflow States, You Might Need a SearchDummy. Workflow states and other exception conditions (e.g., escalated bug or credit hold) are usually the most urgent action items in a CRM system. While they probably show up in a dashboard or report, why not have them available as searchable items? For example, "SLA violations AND New York" could be helpful for a customer support person trying to find the most urgent cases to work on.

If You Can't search for Booleans, You Might Need a SearchDummy. Nobody would ever want to search for the abstract TRUE or FALSE flags, but they would want to search for the semantic consequences of those flags. For example, "Strategic Customer AND Toronto" can be a helpful way to find that Canadian customer you've forgotten the name of.

This trick can also be used to expand the keyword Boolean functions in the CRM search. Adding a NOT string (such as NOTcustomer) into the SearchDummy lets users search for the lack of something, even if your system's search doesn't allow that. "GREATERthan1/1/2000" or "LESSthan 5 items" can be similarly useful expansions.

How Do I Create These Miraculous SearchDummy Fields?

First, let's describe what they are: a string that is populated from the values in one or more fields on a record. Each table (or object) needs to have a separate SearchDummy defined and populated.

While some of the SearchDummy use cases are no-brainers, the more interesting ones need to be designed to properly handle users' expected searches. This is particularly true with the numerical ranges and Booleans, as you want to create only a few well-targeted ones. Create too many, and the search result sets become too large to be useful.

The general checklist (the details vary by CRM vendor):

- Create a SearchDummy field on the table or object in question. You may find it easier to create several of them, one for each approach below.
- Use formula fields to create a text string of the references or items that you want to search for.
- Use workflow functions to populate SearchDummies when certain conditions have been achieved.
- Use triggers or application code to populate key data or conditions across objects.

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 clients are in North America, Europe, Israel, and India, and David has over 25 years experience in high tech, including 10 years at the VP level or above.

Follow everything from CIO.com on Twitter @CIOonline, and the CIO.com Facebook page