

Split Single Address Field into Component Parts

Source: <http://www.tech-archive.net/Archive/Access/microsoft.public.access.queries/2004-04/0571.html>

From: Ted Allen (*anonymous_at_discussions.microsoft.com*)

Date: 04/06/04

Date: Tue, 6 Apr 2004 08:40:13 -0700

Hi Ben,

I'm sure there are numerous ways of attacking this, but I'll describe the way that I dealt with this problem in the past (we had a database of hundreds of thousands of records from a City Billing Agency to deal with).

In my case, and probably in your case, the data format was not uniform so it wasn't possible to do all of the updates at once – so this is what I did.

I added extra fields for all of the parsed data that I wanted to populate, and also one field for the temporary working address.

I started by copying the original address to the working address field using an update query. And, as always, I would make a copy of the database before making any updates just in case.

I then looked at the data and parsed it step by step using update queries.

For each step, I looked at the data and decided what I could pull out by setting conditions (criteria) and using string functions. I would then use the update query to parse part of the address to the appropriate field, and would subtract that part of the string from the working address. Using the working address field in this way allows you to have one field containing only the remaining address pieces to be parsed, while leaving the original address in tact to compare your results to.

For instance, I think I handled the PO Boxes first because they did not fit most of the norms and I wanted to get them out of the way first. First I did a few direct updates on the working address to modify "P.O." "PO" "P. O." etc to a uniform format (this

microsoft.public.access.queries: Split Single Address Field into Component Parts

would be optional – but improves consistency)

Then, I ran an update query to extract the portion of the string containing the P.O. Box info to the new target field (you could have a dedicated PO Box field, or use Street field – I think I used a dedicated field).

This update query would include criteria that the working address is like "P.O. Box*", and the PO Box field would be updated to `left([WorkingAddress],8)` while the working address would be modified to `right([WorkingAddress],len([WorkingAddress])–8)`. Of course, you would have to handle this in multiple passes if the P.O. Box info is not in uniform condition.

Then, I think the next thing that I did was extract the house numbers. For that one, I used the criteria that the leftmost character was numeric, something like:

```
IsNumeric(left([WorkingAddress],1))
```

I would then use `Instr([WorkingAddress]," ",1)` to find the position of the first space within the working address, and use the left and right functions to update the WorkingAddress and the HouseNo fields.

For example HouseNo would be `left([WorkingAddress],Instr([WorkingAddress]," ",1)–1)`. This says to take the left portion of the string before the first space (the one in the Instr function specifies to use text comparison). If you wanted to be extra careful you could set another where condition specifying that the result of this function must be numeric – `IsNumeric(left([WorkingAddress],Instr([WorkingAddress]," ",1)–1)) = True`

I would then repeat this with each portion of the address. Generally, with each step, I would first start with a select query to view the records that would meet the criteria and the calculated values. If I saw some records that wouldn't work right, I would adjust the criteria to exclude them for a separate update, or adjust the formula to work for all if possible.

To get to the main point of your original question, which is how to extract other info than just the left portion, you can use the following functions to extract any piece of the string (Look in Access or VB help for details):

Left() – Extract left portion of string
Right() – Extract right portion of string
Mid() – Extract middle portion of string

microsoft.public.access.queries: Split Single Address Field into Component Parts

Len() – Return Length of string
Instr() – Search for one string within another and return position
InstrRev() – Same as above but from the right
IsNumeric() – True if the entire string being evaluated is numeric
LCase() – convert to lowercase
UCase() – CONVERT TO UPPERCASE
StrConv() – Convert to lower, UPPER, or Title Case

And, I'm sure there are others that I'm not thinking of.

Mostly I find myself using left(), right() and mid() in combination with len() and instr().

I'm not aware of any pre-programmed modules that will do all this for you, but I wouldn't be surprised if they exist – since this is a common task. But, it would be difficult since the data being parsed is often in bad shape, and not in consistent format. We had one database that we received where they had just typed the mailing info in four fields, for lines 1–4. Sometimes the street address would be on line 2, sometimes on line 3, etc. That was ugly.

Hopefully this will help somewhat. I guess the summary of my approach is to look at the data and update as much as I can in chunks according to patterns that I see in the data. The number of updates that it would take would depend on how bad the uniformity of the data is.

Post back if you have other questions.

–Ted Allen

>-----Original Message-----

>Hi,

>I'm working on a database for a client. They want the
>database improved and the info displayed in a different
>format. Short story: at the moment there is one address
>field that contains "17 Some Street". I want to, within
a

>query, split that single field so that I end up with
three

>fields – as in "StrNumber", "StrName", "StrType".

>

>I've worked out how to strip out the street number using
>the Left function, but I can't work out how to do the
rest.

>

>Further problem, in case you're really feeling ambitious,
>is that some of these address fields also contain

microsoft.public.access.queries: Split Single Address Field into Component Parts

>completely different strings, ie. "C/- Some Real Estate"
or
>"PO Box 99". It'd be great to be able to differentiate,
>but I imagine that's a pretty big ask. I don't want to
do
>it manually because there are over 10000 records in this
>table. And the reason I'm using a query is that I'm
trying
>to rebuild the database a little more efficiently by
taking
>a single 70-odd field table and splitting it down into
more
>relevant, related tables.
>
>All help appreciated.
>
>Cheers,
>Ben.
>
>