

## Re: Querying for value that shouldn't be in column?

**Source:**

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.formscoding/2004-02/0140.html>

---

**From:** Marshall Barton (*marshbarton\_at\_wowway.com*)

**Date:** 02/05/04

Date: Wed, 04 Feb 2004 21:34:58 -0600

Tim Rogers wrote:

>I truly appreciate you taking the time to help me with this.  
>Although, I'm not sure I'm following you. I suspect I'm not  
>explaining my problem correctly. The hard-coded string is not in the  
>lookup table at all. So, there is no corresponding ID. It ends up in  
>tblNotes only because a form in the app allows a user to do something  
>that causes the form code to stuff the hard-coded string value into  
>the field (as opposed to one of the lookup values).  
>  
>In other words, the developer displays the lookup values in a combo  
>box, but then adds a particular string value to the combo box's  
>contents. So, the user can choose that string value which has  
>absolutely no connection to the lookup table or any of it's values.  
>So, when I'm trying to search tblNotes for any rows that have that  
>string value (in the field that really should only have the lookup  
>values) I can't get a match. When I look at tblNotes I see rows that  
>do have that hard-coded string displayed, but my query that attempts  
>to match on that displayed string fail.  
>  
>Does this explanation change your the approach you outlined below?  
>I'm just sure I understand what you're telling me.

I'm afraid I'm even more lost than I thought I was!? It sounds like you're permitting users to enter a string into a combo box when the combo box is supposed to return an ID. Either that or you're trying to look up a value that either you know is not there or, if there, is supposed to return the same thing you looked up???

Could you expand on your explanation with some example data and what result you hope to get?

--

Marsh

MVP [MS Access]

>> Tim Rogers wrote:

## microsoft.public.access.formscoding: Re: Querying for value that shouldn't be in column?

```
>> >Well, my problem really is with trying to 'match' the hard-coded
>> >string value in a query. I've actually boiled down my query to simply
>> >find the rows that match on the hard coded string. So, the query
>> >
>> >SELECT tblNotes.NoteID, tblNotes.SiteID
>> >FROM tblNotes
>> >WHERE (((tblNotes.SiteID)=<Some_Hard_Coded_String_Value>));
>> >
>> >should return all rows where SiteID equals the hard-coded value. The
>> >problem is it does not. So, my question is really about how to make
>> >the above query work. tblNotes has the column I described in my post
>> >that, in it's design, has SiteID as a lookup field where the values
>> >come from another table. The query for the lookup pulls back the PK
>> >and a 'friendly' string value. This field is bound to column 1 from
>> >the query (i.e. the PK). But in the form that uses tblNotes the user
>> >can not only select from these lookup values but can also choose a
>> >hard-coded string that the developer coded into the form.
>> >
>> >Since the SiteID table is designed to use a 2-column result set (a PK
>> >and a friendly string value), I believe a "1 column" hard-coded string
>> >value being inserted into this field is causing my grief. I assume
>> >that Access somehow, behind the scenes, must be resolving the 'bind to
>> >column 1, but display column 2' design with the hard-coded string
>> >value. So, when I search for that hard-coded string value in a query,
>> >I don't find it. I assume Access has put something in for 'column 1'
>> >or maybe 'column 2' so it can treat all values in this field the same.
>> >
>> >So, how can I do a query and search for this hard-coded value in this
>> >field?
>>
>>
>Marshall Barton wrote:
>> This kind of confusion is causing me to hate those bleeping
>> lookup fields more and more everyday. Arrrgggghhh
>>
>> But, at least, you do seem to understand the core of the
>> issue. The hard coded friendly string must either be
>> translated to it's corresponding ID number, which would then
>> be used in the query. This might be doable with a DLookup
>> sort of like:
>>
>> lngstringID = DLookup("idfield", "thelookuptable",
>> "friendlystringfield="" & hardcodedfriendlystring & """)
>>
>> then the query would be more like:
>>
>> SELECT tblNotes.NoteID, tblNotes.SiteID
>> FROM tblNotes
>> WHERE thelookuptable.SiteID=<the value of lngstringID>
>>
>> or the query must include the friendly string so the Where
>> clause can compare to it. The SQL would be something like:
>>
>> SELECT tblNotes.NoteID, tblNotes.SiteID
>> FROM tblNotes INNER JOIN thelookuptable
>> ON tblNotes.SiteID = thelookuptable.SiteID
>> WHERE thelookuptable.friendlystringfield =
>> "<Some_Hard_Coded_String_Value>"
```