

Einfacher Listenfeldfilter

Manchmal möchte Sie einen Filter auf die einfache Art setzen – ohne VBA und den Einsatz von Ereignisprozeduren. Das gelingt unter Access sehr leicht, aber nicht ohne Tücken – und auch nicht komplett ohne den Einsatz von VBA: So kann es zum Beispiel sein, dass das Listenfeld keine Daten anzeigt, wenn das für die Eingabe des Vergleichswertes verwendete Textfeld leer ist. Wir schauen uns einen einfachen Listenfeldfilter anhand eines Textfeldes an und zeigen, wie Sie die kleinen Hindernisse umschiffen.

Beispieldatenbank

Die Beispiele dieses Artikels finden Sie in der Datenbank **1806_EinfacherListenfeldfilter.accdb**.

Vorbereitung

Das Formular soll ein Textfeld namens **txtSucheNachFirma** enthalten sowie ein Listenfeld namens **IstKunden**, das zu Beginn alle Datensätze der Tabelle **tblKunden** anzeigt (siehe Bild 1). Dem Listenfeld weisen wir als Datensatzherkunft die folgende Abfrage zu, um die Spalten korrekt auszurichten:

```
SELECT tblKunden.KundeID, tblKunden.KundenCode, tblKunden.Firma, tblKunden.Kontaktperson FROM tblKunden;
```

Dabei wollen hier nur die Spalten **KundeID**, **Kundencode**, **Firma** und **Kontaktperson** anzeigen. Damit die Spalte **KundeID** nur als gebundene Spalte verwendet, aber nicht angezeigt wird, stellen wir die Eigenschaft **Spaltenanzahl** auf 4 und **Spaltenbreiten** auf 0cm ein. Dadurch wird die erste Spalte ausgeblendet und die übrigen Spalten teilen sich den verbleibenden Platz.

Das Ergebnis sieht in der Formularansicht nun wie in Bild 2 aus. Hier haben wir auch noch die Eigenschaften **Navigationsschaltflächen**, **Datensatzmarkierer**, **Bildlaufleisten** und **Trennlinien** auf **Nein** sowie **Automatisch zentrieren** auf **Ja** eingestellt. Hier zeigt sich, dass wir die Breiten der Spalten doch manuell einstellen müssen, um ein brauchbares Ergebnis zu erhalten. Wir stellen die Eigenschaft **Spaltenbreiten**

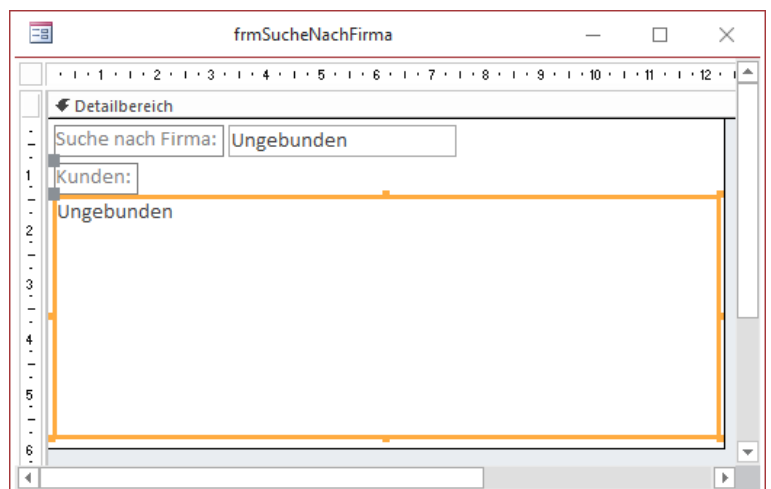


Bild 1: Entwurf des Beispielformulars

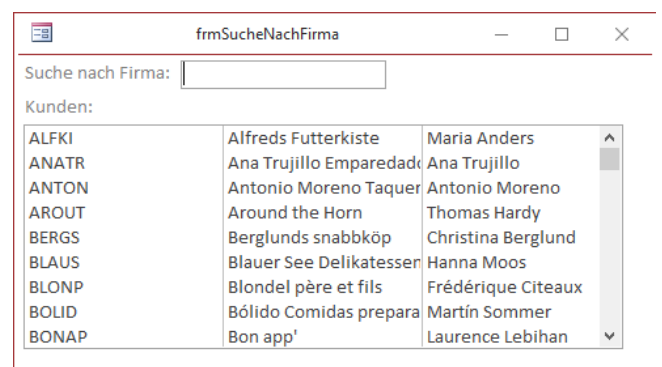


Bild 2: Formularansicht des Beispielformulars

auf **0cm;2cm** ein. Damit wird die Spalte mit dem Kundencode schmaler und die beiden übrigen Spalten teilen sich den verbleibenden Platz.

Filter einbauen

Den Filter bauen wir nun direkt in die Abfrage ein, die wir als Datensatzquelle des Listenfeldes angeben

haben. Diese erhält nun eine **WHERE**-Bedingung, mit der wir das Feld **Firma** nach dem Wert des Textfeldes **txtSucheNachFirma** filtern und sieht anschließend wie in Bild 3 aus.

Wenn wir das Formular nun in der Formularansicht öffnen, zeigt das Listenfeld gar keine Datensätze an. Das ist nicht überraschend, denn nach der Definition der Datensatzherkunft sollen nun alle Datensätze angezeigt werden, deren Feld **Firma** leer ist (siehe Bild 4).

Keine Ergebnisse ohne Aktualisieren

Wenn wir einen Suchbegriff eingeben wie etwa **A***, von den wir wissen, dass er einige Ergebnisse liefern sollte, erhalten wir immer noch keine Ergebnisse. Der Grund ist, dass sich das Listenfeld nicht automatisch aktualisiert. Dazu müssen Sie den Fokus auf das Listenfeld verschieben und die Taste **F5** zum Aktualisieren betätigen. Das ist natürlich kein Ablauf, den Sie dem Benutzer der Anwendung zumuten wollen. Also werden wir wohl oder übel doch eine kleine VBA-Ereignisprozedur implementieren müssen, die nach dem Aktualisieren des Inhalts unseres Such-Textfeldes den Inhalt des Listenfeldes aktualisiert. Dazu hinterlegen wir die folgende Ereignisprozedur für das Ereignis **Nach Aktualisierung** des Textfeldes:

```
Private Sub txtSucheNachFirma_AfterUpdate()  
    Me!lstKunden.Requery  
End Sub
```

Danach zeigt das Listenfeld dann auch die gewünschten Datensätze an.

Alle Einträge für leeres Suchfeld

Wie können wir nun ohne den Einsatz weiterer VBA-Anweisungen dafür sorgen, dass das Listenfeld auch

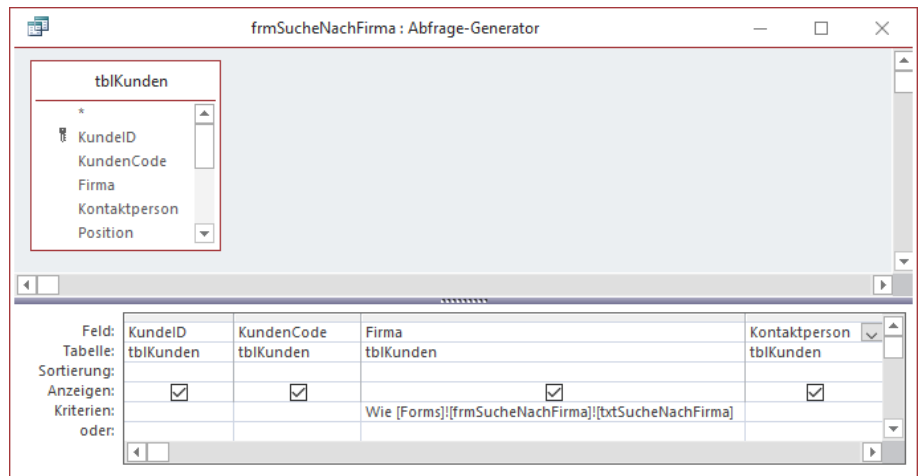


Bild 3: Datensatzherkunft des Listenfeldes mit Kriterium

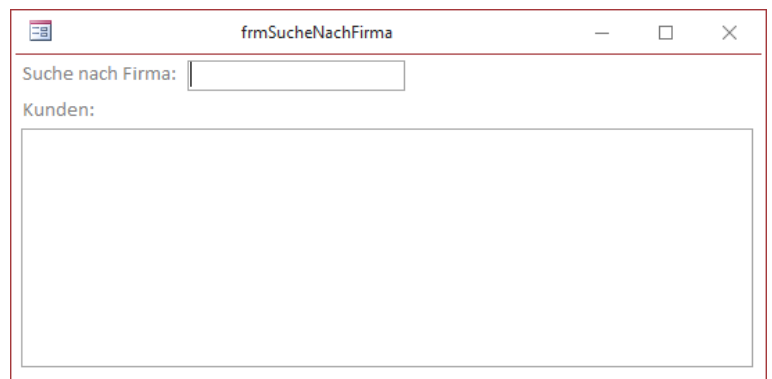


Bild 4: Leeres Suchergebnis ohne Eingabe eines Vergleichswertes

dann alle Datensätze anzeigt, wenn der Benutzer das Suchfeld leert beziehungsweise wenn es nach dem Anzeigen des Formulars noch leer ist?

Das können wir auch erreichen, indem wir die Datensatzherkunft des Listenfeldes weiter anpassen. Dazu stellen wir den Vergleichswert für das Feld **Firma** auf den folgenden Ausdruck ein:

```
Wie Nz([Forms]![frmSucheNachFirma]![txtSucheNachFirma]; "*")
```

Die Abfrage sieht danach wie in Bild 5 aus.

Suche mit Abfrage per VBA

Einen Schritt weiter gehen Sie, wenn Sie beim Öffnen des Formulars einen Abfrageausdruck für das Listenfeld **IstKunden** hinterlegen, der immer alle Datensätze

dieser Tabelle anzeigt – also etwa die folgende Abfrage, welche der vorherigen Abfrage ohne Verwendung der **WHERE**-Klausel entspricht:

```
SELECT KundeID, Kunden-
Code, Firma, Kontaktperson
FROM tblKunden
```

Bei Öffnen zeigt das Listenfeld dann alle Datensätze an. Aber wie sorgen wir dann ohne ein Kriterium in dieser Abfrage dafür, dass die Datensätze nach der Eingabe in das Textfeld `txtSucheNachFirma` gefiltert werden? Dazu weisen Sie in der Ereignisprozedur `txtSucheNachFirma_AfterUpdate` die komplette Datensatzquelle neu zu:

```
Private Sub txtSucheNachFirma_AfterUpdate()
    Me!lstKunden.RowSource = "SELECT KundeID, 7
        Kundencode, Firma, Kontaktperson FROM 7
        tblKunden WHERE Firma LIKE '' 7
        & Nz(Forms!frmSucheNachFirmaVBA!7
        txtSucheNachFirma, "") & ""
End Sub
```

Auch hier verwenden wir wieder die **Nz**-Funktion, um zu untersuchen, ob der Benutzer überhaupt einen Ausdruck für das Textfeld `txtSucheNachFirma` eingegeben hat. Anderenfalls soll die **Nz**-Funktion einfach das Sternchen als Zeichenkette zurückliefern ("").

```
Private Sub txtSucheNachFirma_AfterUpdate()
    If Not IsNull(Me!txtSucheNachFirma) Then
        Me!lstKunden.RowSource = "SELECT KundeID, Kundencode, Firma, Kontaktperson FROM tblKunden WHERE Firma LIKE '' _
            & Me!txtSucheNachFirma & ""
    Else
        Me!lstKunden.RowSource = "SELECT KundeID, Kundencode, Firma, Kontaktperson FROM tblKunden"
    End If
End Sub
```

Listing 1: Diese Prozedur öffnet ein neues Formular oder zeigt dieses an, wenn es bereits geöffnet ist.

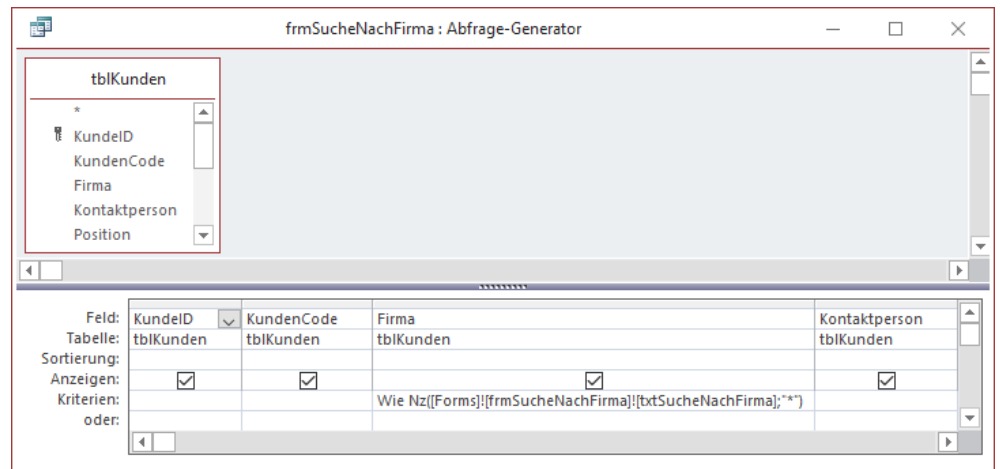


Bild 5: Suchkriterium zur Anzeige aller Datensätze bei leerem Suchfeld

Dabei können wir den Ausdruck `Forms!frmSucheNachFirmaVBA` noch durch `Me` ersetzen, denn innerhalb des Klassenmoduls eines Formulars können Sie mit diesem Schlüsselwort auf das Formular selbst zugreifen:

```
Private Sub txtSucheNachFirma_AfterUpdate()
    Me!lstKunden.RowSource = "SELECT KundeID, 7
        Kundencode, Firma, Kontaktperson FROM tblKunden 7
        WHERE Firma LIKE '' & Nz(Me!txtSucheNachFirma, 7
        "") & ""
End Sub
```

Sie können die **Nz**-Funktion auch noch wie in Listing 1 nach einer Untersuchung des Inhalts von `txtSucheNachFirma` durch eine **If...Then**-Bedingung ersetzen. Die Bedingung prüft, ob das Textfeld leer ist. Falls nicht, wird die Abfrage mit **WHERE**-Bedingung verwendet, anderenfalls ohne diese Bedingung.