



Kunden per Bezeichnung verwalten

Beim Umgang mit Kundendaten benötigen Sie oft eine allgemeine Bezeichnung eines Kunden. Diese soll beispielsweise als Überschrift im Formular zur Verwaltung der Stammdaten eines Kunden dargestellt werden oder als Wert zur Auswahl in einem Kombinationsfeld angeboten werden. Praktisch wäre es, diesen Ausdruck etwa in einem Abfragefeld in der Form <Firma> (<Nachname>, <Vorname>) oder ähnlich abzubilden. Das gelingt jedoch nicht so leicht, weil ein Kunde mal mit, mal ohne Firma kommt. Und was geschieht überhaupt, wenn zwei Kunden den gleichen Namen haben? Diese und weitere Fragen beantwortet dieser Artikel.

Beispieldatenbank

Die Beispiele dieses Artikels finden Sie in der Datenbank **1111_KundePerBezeichnung.mdb**.

Kundenbezeichnung erwünscht

Ein Einsatzzweck für eine Kundenbezeichnung, die sich aus der Firma des Kunden und dem Namen des Ansprechpartners zusammensetzt, wäre etwa die Darstellung in einem Kombinationsfeld zur Auswahl eines Kunden etwa in einem Bestellformular.

Normalerweise werden solche Kombinationsfelder immer mit der Kundentabelle als Datensatzherkunft und dem Nach- und dem Vornamen als anzuzeigendem Wert dargestellt.

Dabei fällt allerdings die Firmenbezeichnung weg. Wenn man die Firmenbezeichnung, die aber in der Regel nur bei einem Teil der Kunden vorliegt, hinzufügen möchte – etwa in der Form **Minhorst, André (Redaktionsbüro Minhorst)** oder **Redaktionsbüro Minhorst (Minhorst, André)**, muss man einen Wenn-Ausdruck in der Abfrage verwenden.

Andersherum kann es auch sein, dass zu einem Kunden gar kein Ansprechpartner vorliegt, sondern nur der Firmenname. Gegebenenfalls möchten Sie auch die Kundennummer in den Ausdruck für ein Kombinationsfeld oder für die Titelzeile des Detailformulars integrieren.

Richtig unpraktisch wird es, wenn zwei Kunden den gleichen Namen haben. Dann können Sie, wenn nur Vorname und Nachname angezeigt werden, anhand des Eintrags im Kombinationsfeld nicht auf die Schnelle entscheiden, ob Sie auch den richtigen Kunden auswählen.

Und da in diesem Fall der Ausdruck dynamisch etwa durch Feld einer Abfrage ermittelt wird, lässt sich auch nichts an diesem Ausdruck ändern – es sei

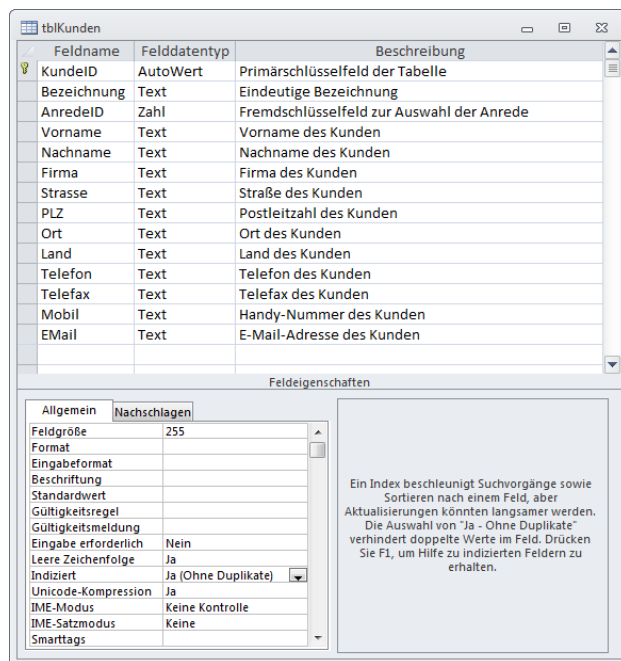


Bild 1: Kundentabelle mit Bezeichnungsfeld

denn, Sie ändern die zugrunde liegenden Daten. Das ist wiederum unmöglich – es sei denn, Sie möchten die verschiedenen Michael Müllers in Ihrem Kundenstamm mit **Müller1**, **Müller2** und **Müller3** betiteln. Spätestens beim Erstellen von Schriftverkehr auf Basis dieser Daten kommt es zu Problemen.

Separates Bezeichnungsfeld

Die Tabelle **tblKunden** der Beispieldatenbank verfügt neben den üblichen Feldern über ein Feld namens **Bezeichnung**.

Dieses Feld soll eine eindeutige Bezeichnung eines jeden Kunden gewährleisten und wurde deshalb mit einem eindeutigen Index versehen – dies erreichen Sie beispielsweise durch das Einstellen der Eigenschaft **Indiziert** auf den Wert **Ja (Ohne Duplikate)**. Bild 1 zeigt den Entwurf der Tabelle mit dem Feld **Bezeichnung**.



Bezeichnung automatisch anpassen

Zur Anzeige der Daten der Tabelle **tblKunden** erstellen Sie ein Formular namens **frmKunden**, das die Tabelle **tblKunden** als Datenherkunft verwendet. Ziehen Sie alle Felder der Tabelle in den Detailbereich des Formularentwurfs und ordnen Sie die Felder nach Wunsch an – beispielsweise so wie in Bild 2.

Typischerweise liegen schon Kundendaten vor, für die Sie die Bezeichnung nachträglich hinzufügen müssen, für neue Datensätze soll die Bezeichnung beim Speichern des Datensatzes eingetragen werden.

Wie aber soll die Bezeichnung nun aussehen? Dies ist letztlich egal, die paar Zeilen Code, welche die Bezeichnung zusammenstellen, können Sie leicht an die eigenen Bedürfnisse anpassen.

Wir entscheiden uns für das folgende Schema:

```
<KundeID> - <Firma> (<Nachname>, <Vorname>)
```

Für neue Datensätze soll die Bezeichnung zunächst beim Speichern des Datensatzes angelegt werden.

Damit noch nachträgliche Änderungen notwendig sind, bringen wir die notwendigen Codezeilen in der Ereignisprozedur unter, die durch das Ereignis Vor Aktualisierung ausgelöst wird.

Legen Sie diese Prozedur zunächst an, indem Sie in die Entwurfsansicht des Formulars wechseln und im Eigenschaftsfenster für den Eintrag **Vor Aktualisierung** den Wert **[Ereignisprozedur]** auswählen. Danach klicken Sie auf die Schaltfläche mit den drei Punkten (...) und füllen die nun im VBA-Editor angezeigte Prozedur wie folgt:

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
    Me!Bezeichnung = Me!KundeID & " - " _
        & Me!Firma & " (" & Me!Nachname & ", " _
        & Me!Vorname & ")"
End Sub
```

Wenn Sie nun einen neuen Datensatz anlegen und diesen speichern (entweder durch Verlassen des Datensatzes, durch Betätigen der Tastenkombination **Strg + S** oder durch einen Mausklick auf den Datensatzmarkierer), wird die Bezeichnung wie in Bild 3 angepasst. Die einzige Anweisung dieser Ereignisprozedur fügt den Inhalt einiger Felder und einige

The screenshot shows a form titled 'frmKunden' with the following fields and values:

- Kunde-ID: 1
- Bezeichnung: (empty)
- Anrede: Frau
- Vorname: Miriam
- Nachname: Dill
- Firma: Scharf AG
- Strasse: Neubastraße 44
- PLZ: 90427
- Ort: Nürnberg
- Land: Deutschland
- Telefon: 0911/023796
- Telefax: 0911/0672534
- Mobil: 030/935705
- E-Mail: miriam@dill.de

At the bottom, there is a status bar with 'Datensatz: 1 von 100', 'Kein Filter', and a 'Suchen' button.

Bild 2: Formular zur Anzeige der Kundendaten

The screenshot shows the same 'frmKunden' form, but the 'Bezeichnung' field now contains the text '1 - Scharf AG (Dille, Miriam)'. The other fields remain the same as in Bild 2.

Bild 3: Kunde mit angepasster Bezeichnung

Zeichen wie das Minuszeichen und die öffnende und schließende Klammer zusammen.

Bestehende Daten mit Bezeichnung versehen

Wenn Sie bereits bestehende Kundendaten um eine Bezeichnung ergänzen wollen, verwenden Sie am einfachsten eine Aktualisierungsabfrage. Diese verwendet etwa den gleichen Ausdruck wie die VBA-Prozedur von oben und sieht wie in Bild 4 aus. Nach dem Ausführen der Abfrage erhalten Sie das Ergebnis aus Bild 5 – natürlich nur im Optimalfall, nämlich wenn alle Felder der Datensätze gefüllt sind (dazu später mehr).

Bezeichnung einsetzen

Sie können die Bezeichnung nun beispielsweise in der Titelleiste des Detailfensters zur Anzeige von Kunden einsetzen. Damit diese immer die Daten zum



aktuellen Kunden anzeigt, legen Sie eine Ereignisprozedur an, die beim Anzeigen eines Datensatzes ausgelöst wird und die dementsprechend **Beim Anzeigen** heißt. Die Ereignisprozedur enthält nur eine einzige Zeile, die wie folgt aussieht:

```
Private Sub Form_Current()
    Me.Caption = _
        Me!Bezeichnung
End Sub
```

Das Formular zeigt die Bezeichnung nun in der Titelleiste an (siehe Bild 6).

Bezeichnung in Kombinationsfeldern

Außerdem können Sie die Bezeichnung nutzen, wenn Sie einen Kunden zu irgendeinem Zweck per Kombinationsfeld auswählen möchten. Diesem weisen Sie dann als Datensatzherkunft die Abfrage aus Bild 7 zu.

Stellen Sie die Eigenschaften **Spaltenanzahl** und **Spaltenbreiten** auf die Werte **2** und **0cm** ein, damit das Formular das gebundene Feld, also **KundeID**, ausblendet und nur den Inhalt des Feldes **Bezeichnung** anzeigt.

In der Abfrage haben wir eine Sortierung nach dem Firmennamen eingestellt. Dies können Sie natürlich nach Ihren eigenen Vorstellungen anpassen.

Danach können Sie die Kunden mit einem Kombinationsfeld wie in Bild 8 auswählen.

Bezeichnungen durchsuchen

Eine weitere Möglichkeit ist es, die Bezeichnung nach bestimmten Werten zu durch-

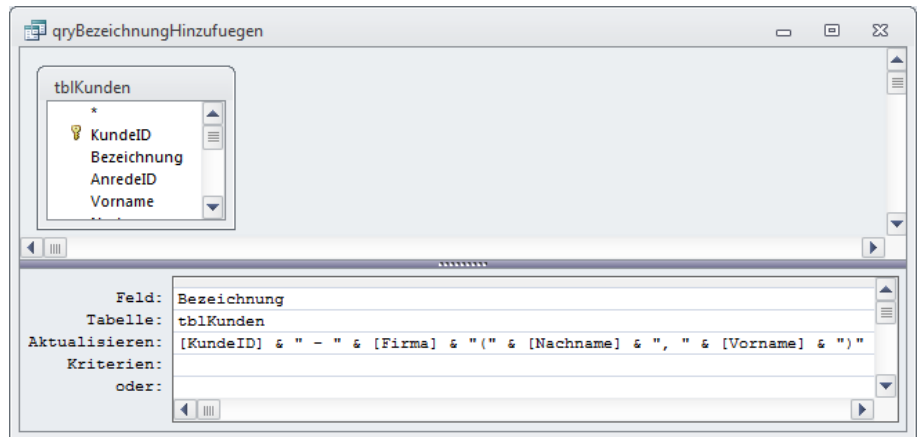


Bild 4: Aktualisieren aller Bezeichnungen der Kundentabelle

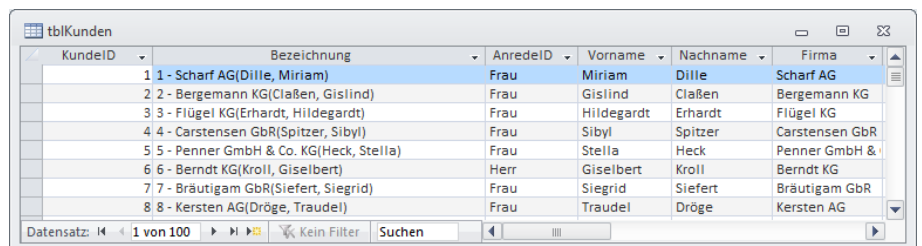


Bild 5: Kunden mit Bezeichnung

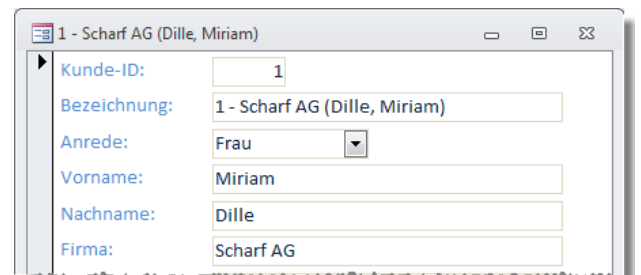


Bild 6: Anzeige der Bezeichnung im Formulartitel

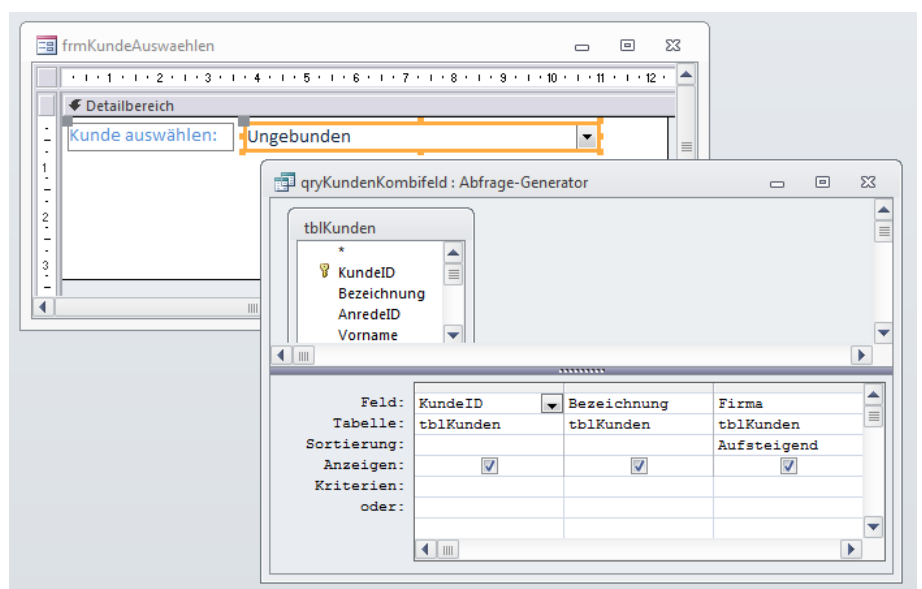


Bild 7: Kombinationsfeld mit Kundenbezeichnungen füllen



suchen – in unserer Version sind ja bereits einige Daten wie Kundennummer, Vorname, Nachname oder Firma enthalten. Mehr dazu erfahren Sie im Artikel **Suche nach Feldinhalten**.

Finetuning

Mit der aktuellen Konstellation ergeben sich mitunter die folgenden Probleme:

- Es fehlen Inhalte wie **Vorname** und **Nachname** oder **Firma**. Daraus resultieren dann Ausdrücke wie **1 - <Firma> (,)** oder **1- (<Nachname>, <Vorname>)**, das heißt, die Bezeichnung wird ohne Rücksicht auf leere Bereiche zusammengestellt.
- Es gibt noch keine Behandlung von Fällen, in denen eine Bezeichnung zweimal vorkommt. Dies ist in unserer Konstellation unmöglich, da ja der Primärschlüsselwert eingebunden ist. Daher stellen wir später eine Variante ohne Primärschlüssel vor.
- Manuelle Änderungen an der Bezeichnung werden beim nächsten Speichern des Datensatzes automatisch wieder überschrieben.

Diese kleinen Schwachstellen bügeln wir noch aus. Als erstes sollen die Daten entsprechend der vorliegenden Daten optimal zusammengestellt werden.

Wenn etwa keine Firma vorhanden ist, sollen Vorname und Nachname statt einer Firma eingesetzt werden, wenn kein Vor- und Nachname vorliegt, wird der Klammern-Teil schlicht weggelassen.

Der folgende Ansatz berücksichtigt, dass eine Bezeichnung nur beim ersten Anlegen des Benutzers automatisch mit Änderung der beteiligten Felder angepasst wird, danach müssen Änderungen durch einen Klick auf eine entsprechende Schaltfläche vorgenommen werden.

Auch wenn die Bezeichnung beim Anlegen des Kundendatensatzes einmal manuell angepasst wurde, sollen keine automatisch Änderungen mehr stattfinden.

Damit dies nicht geschieht, legen Sie im Klassenmodul des Formulars eine Variable namens **bolAutomatischAendern** an:

```
Dim bolAutomatischAendern As Boolean
```

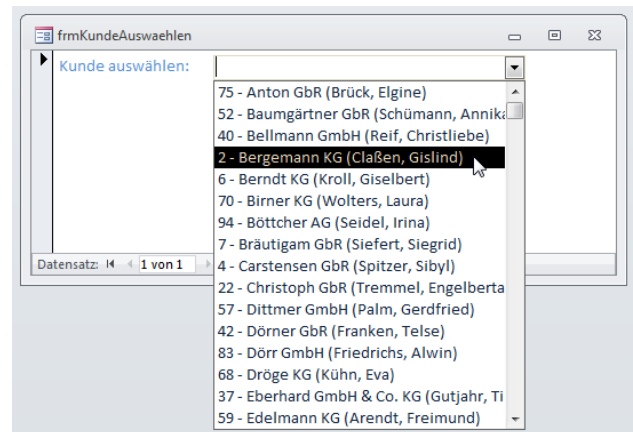


Bild 8: Kundenauswahl per Kombinationsfeld

Wenn ein Datensatz angezeigt wird, prüft die Prozedur, die durch das Ereignis **Beim Anzeigen** ausgelöst wird, ob die Bezeichnung noch leer ist. Falls ja, können Änderungen an den drei Feldern **Vorname**, **Nachname** und **Firma** den Wert des Feldes **Bezeichnung** beeinflussen:

```
Private Sub Form_Current()  
    bolAutomatischAendern = IsNull(Me!Bezeichnung)  
    Me.Caption = Nz(Me!Bezeichnung, "Neuer Kunde")  
End Sub
```

Auch wenn der Benutzer beim Anlegen erstmals manuelle Änderungen am Feld **Bezeichnung** vorgenommen hat, sollen Änderungen an den übrigen Feldern keine automatische Aktualisierung des Feldes mehr auslösen. Das manuelle Ändern des Wertes im Feld **Bezeichnung** löst das Ereignis **Bei Änderung** aus, hinter der sich folgende Prozedur verbirgt:

```
Private Sub Bezeichnung_Dirty(Cancel As Integer)  
    bolAutomatischAendern = False  
End Sub
```

Durch das Setzen von **bolAutomatischAendern** auf **False** wird die automatische Änderung deaktiviert.

Die Änderung wird grundsätzlich ausgelöst, wenn der Benutzer den Vornamen, den Nachnamen oder die Firma ändert oder wenn er explizit auf die Schaltfläche **cmdBezeichnungAnpassen** klickt. Die Änderungen der Feldinhalte lösen jeweils das Ereignis **Nach Aktualisierung** der jeweiligen Steuerelemente aus, was die folgenden Ereignisprozeduren aufruft:

```
Private Sub Firma_AfterUpdate()  
    BezeichnungZusammenstellen  
End Sub
```



```
Private Sub Nachname_AfterUpdate()
    BezeichnungZusammenstellen
End Sub
```

```
Private Sub Vorname_AfterUpdate()
    BezeichnungZusammenstellen
End Sub
```

Ein Klick auf die Schaltfläche **cmdBezeichnungAnpassen** löst ebenfalls eine Ereignisprozedur mit der einzigen Anweisung **BezeichnungZusammenstellen** aus:

```
Private Sub cmdBezeichnungAnpassen_Click()
    BezeichnungZusammenstellen
End Sub
```

Die von diesen vier Prozeduren aufgerufene Routine **BezeichnungZusammenstellen** erledigt die Hauptarbeit (siehe Listing 1). Sie prüft zunächst, ob die Variable **bolAutomatischAendern** den Wert **True** hat. Nur in diesem Fall werden die innerhalb der **If...Then**-Bedingung enthaltenen Anweisungen ausgeführt. Ist dies der Fall, beginnt die Prozedur, den Bezeichnungs-Ausdrucks in einer Variablen namens **strBezeichnung** zusammenzustellen. Als erstes fügt sie dabei den Wert des Feldes **KundeID** hinzu.

Danach erfolgt eine weitere Prüfung. Diese untersucht, ob das Feld **Firma** bereits einen Wert enthält. Falls ja, wird der Inhalt der Variablen **strBezeichnung** um ein Minus-Zeichen und den Wert des Feldes

```
Private Sub BezeichnungZusammenstellen()
    Dim strBezeichnung As String
    If bolAutomatischAendern Then
        strBezeichnung = Me!KundeID
        If Not IsNull(Me!Firma) Then
            strBezeichnung = strBezeichnung & " - " & Me!Firma
            If Len(Nz(Me!Nachname)) + Len(Nz(Me!Vorname)) > 0 Then
                strBezeichnung = strBezeichnung & " (" & Nz(Me!Nachname) & ", " & Nz(Me!Vorname) & ")"
            End If
        Else
            If Len(Nz(Me!Nachname)) + Len(Nz(Me!Vorname)) > 0 Then
                strBezeichnung = strBezeichnung & " - " & Nz(Me!Vorname) & " " & Nz(Me!Nachname) & " (" & Nz(Me!Nachname) & ", " & Nz(Me!Vorname) & ")"
            End If
        End If
        Me!Bezeichnung = strBezeichnung
        Me.Caption = strBezeichnung
    End If
End Function
```

Listing 1: Kundenauswahl per Kombinationsfeld

Bild 9: Bezeichnung und Titel automatisch anpassen

Firma erweitert. Eine weitere Prüfung ermittelt, ob eines der beiden Felder **Vorname** oder **Nachname** gefüllt ist. Dabei kommt zunächst die **Nz**-Funktion zum Einsatz, die prüft, ob das jeweilige Feld überhaupt einen Wert enthält und die sonst eine leere Zeichenkette statt des Werts **Null** zurückgibt.

Die **Len**-Funktion ermittelt die Länge der beiden Zeichenketten und addiert diese. Wenn mindestens eines der beiden Felder einen Wert enthält, ergibt dies eine Zahl größer als **0** – und in diesem Fall wird der Ausdruck **<Nachname>, <Vorname>** an die bestehende Zeichenkette in **strBezeichnung** angehängt.

Wenn die übergeordnete **If...Then**-Bedingung ermittelt, dass das Feld **Firma** leer ist, wird der **Else**-Teil dieser Bedingung angesteuert. Dieser prüft wiederum, ob **Vorname** oder **Nachname** bereits einen Wert enthalten und hängt dann einen Ausdruck wie **<Vorname> <Nachname> (<Nachname>, <Vorname>)** an die bereits vorhandene Kundennummer in **strBezeichnung** an.

Der resultierende Ausdruck wird wiederum in das Feld **Bezeichnung** eingetragen und als Wert der Eigenschaft **Caption** des Formulars festgelegt, wodurch der Ausdruck in der Titelleiste des Formulars erscheint (siehe Bild 9).

Doppelte Bezeichnungen vermeiden

Einer der wichtigsten Gründe, eine Bezeichnung aus Werten wie **Firma**, **Vorname** oder **Nachname** zusammenzustellen, ist die Anzeige eindeutiger Werte in Kombinati-



onsfeldern und anderen Listen. Da ist es natürlich sinnvoll, bei der Eingabe auch sicherzustellen, dass die Bezeichnung keinen Wert annimmt, der bereits vorhanden ist. Am einfachsten ist es, den Wert beim Speichern des Datensatzes zu prüfen und gegebenenfalls eine Meldung wie in Bild 10 anzuzeigen, um den Benutzer auf einen doppelt vorkommenden Wert hinzuweisen.

Nach dem Bestätigen der Meldung soll der Fokus auf das Feld **Bezeichnung** verschoben werden, damit der Benutzer den Wert anpassen kann.

Ein Beispiel für diese Technik finden Sie im Formular **frmKunden_FirmaVornameNachname**. Hier wird die Kundennummer nicht in die Erzeugung der Bezeichnung einbezogen, da sonst niemals zwei Einträge den gleichen Wert im Feld **Bezeichnung** erhalten könnten.

Wenn eine im Datenmodell festgelegte Restriktion bei der Eingabe von Daten in ein Formular verletzt wird, zeigt Access eine entsprechende Meldung an. Diese können Sie nur unterbinden und durch eine eigene Regel ersetzen, wenn Sie eine Prozedur anlegen, die durch das Ereignis **Bei Fehler** des Formulars ausgelöst wird. In diesem Fall sieht die Prozedur wie folgt aus:

```
Private Sub Form_Error(DataErr As _
    Integer, Response As Integer)
    If DataErr = 3022 Then
        MsgBox "Die Bezeichnung ist " & _
            "bereits vorhanden."
        Me!Bezeichnung.SetFocus
        Response = acDataErrContinue
    End If
End Sub
```

Die Prozedur prüft, ob der Parameter **DataErr**, der die Fehlernummer übergibt, den Wert **3022** liefert. Dieser Fehler tritt auf, wenn der Benutzer versucht, einen Datensatz zu speichern, der mindestens in einem Feld einen eindeutigen Schlüssel verletzt.

In diesem Fall wird eine entsprechende **MsgBox**-Anweisung zum Anzeigen der Meldung aufgerufen. Die Methode **SetFocus** verschiebt den Fokus auf das auslösende Feld. Außerdem sorgt die Prozedur mit

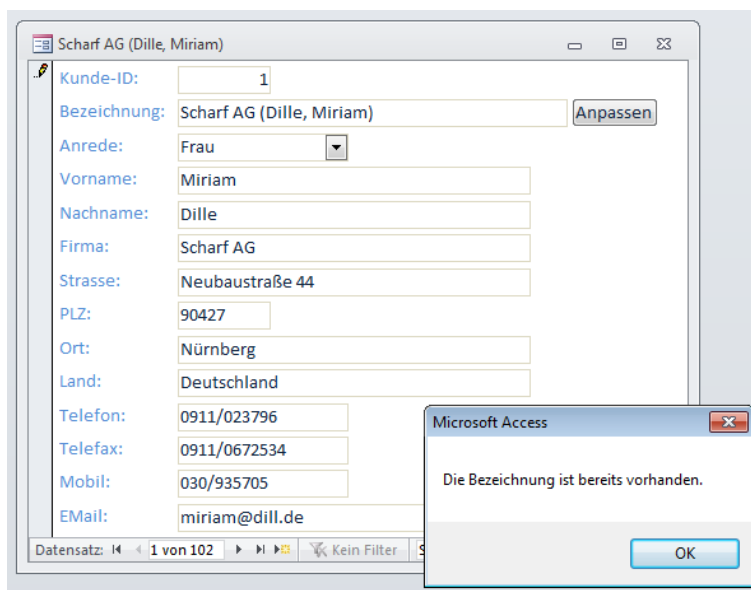


Bild 10: Diese Meldung erscheint, wenn eine Bezeichnung bereits vorhanden ist.

dem Wert **acDataErrContinue** für den Parameter **Response** dafür, dass die eingebaute Fehlermeldung unterdrückt wird.

Zusammenfassung und Ausblick

Eine Bezeichnung eines Kunden, die aus Elementen wie **Kundennummer**, **Firma**, **Vorname** oder **Nachname** besteht, mag unter Umständen redundante Daten enthalten, kann jedoch die Anwendung vereinfachen. Sie können den Inhalt eines solchen Feldes leicht in Titelzeilen, Kombinationsfeldern und nicht zuletzt zum schnellen Durchsuchen von Daten einsetzen.

Wie Letzteres geschieht, erfahren Sie im Artikel **Suche nach Feldinhalten**.

Das Formular **frmKunden** der Beispieldatenbank enthält die hier beschriebene Variante. Unter **frmKunden_Sofortaktualisierung** finden Sie eine Variante, bei der die Änderung der drei Felder **Firma**, **Vorname** und **Nachname** sich beim Eintippen eines jeden einzelnen Buchstaben auf den Wert des Feldes **Bezeichnung** niederschlägt – aber auch nur solange, bis der Benutzer diesen Wert einmal manuell ändert beziehungsweise der Datensatz einmal gespeichert wurde.