

Access-Daten nach Excel verknüpfen

Wir haben uns bereits in verschiedenen Artikeln angesehen, wie Sie von Access aus auf die Daten einer Excel-Datei zugreifen können – ob per VBA oder auch per Verknüpfung oder Import. In diesem Artikel drehen wir den Spieß nun einmal um und zeigen, wie Sie die Daten einer Access-Tabelle in einer Excel-Tabelle einbinden und diese auf dem aktuellen Stand halten. Dabei schauen wir uns sowohl an, wie Sie dies von Excel aus erledigen, als auch wie Sie das Herstellen und Aktualisieren der Verknüpfung von Access aus erledigen.

Beispieldatenbank

Die Beispiele dieses Artikels finden Sie in der Datenbank **1902_Access-UndExcel.zip**.

Verknüpfung von Excel aus initiieren

Wenn Sie die Verknüpfung von Excel aus einrichten wollen, legen Sie zunächst eine neue Excel-Datei über die Benutzeroberfläche von Excel an.

Dazu wählen Sie den Ribbon-Eintrag **Daten|Daten abrufen und transformieren|Daten abrufen|Aus Datenbank|Aus Microsoft Access-Datenbank** aus (siehe Bild 1).

Im nun erscheinenden **Daten importieren**-Dialog wählen Sie die Access-Datenbank aus, welche die Tabelle oder Abfrage enthält, deren Daten Sie in Excel anzeigen wollen.

Danach erscheint ein Dialog namens **Navigator**, der auf der linken Seite alle Tabellen und Abfragen der Quelldatenbank anzeigt und auf der rechten Seite gleich die Daten der aktuell ausgewählten Datenquelle liefert (siehe Bild 2).

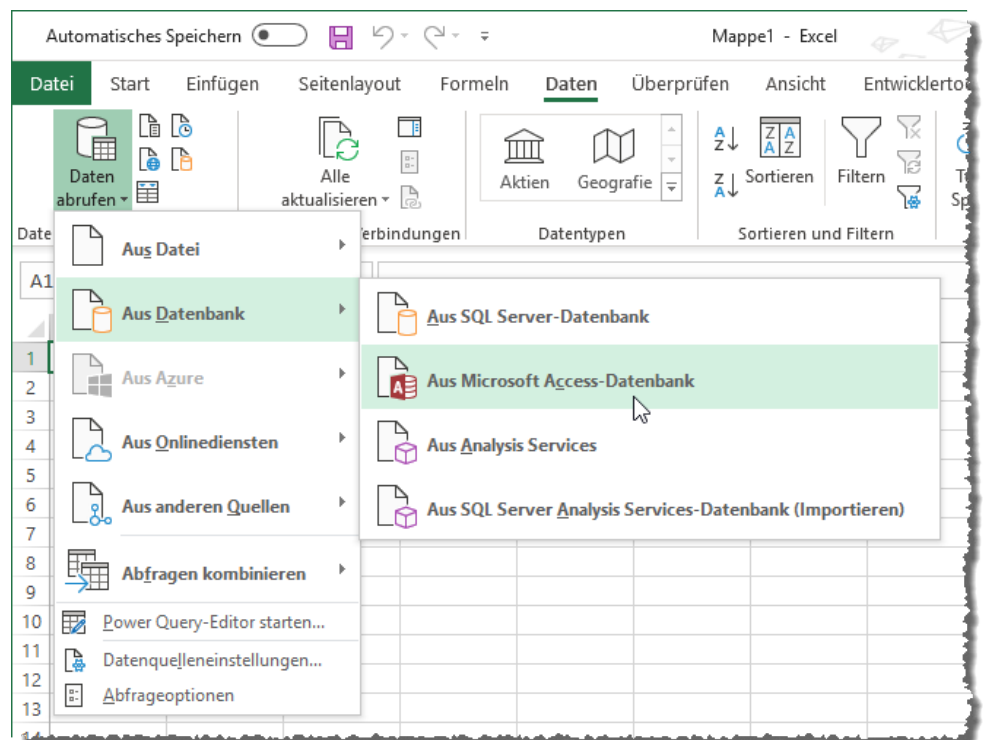


Bild 1: Abrufen von Daten aus einer Access-Datenbank

Klicken Sie in diesem Dialog auf die Schaltfläche **Laden**, nimmt sich Excel ein paar Sekunden Zeit und präsentiert die Daten aus der ausgewählten Quelltable ordentlich formatiert in einer neuen Excel-Tabelle (siehe Bild 3).

Daten aktualisieren

Wenn Sie nun einen Datensatz in der Access-Tabelle anpassen oder einen Datensatz hinzufügen oder löschen, wollen Sie die Änderungen auch in der Excel-Instanz sehen. Dazu klicken Sie nach dem Durchführen der Änderungen in Access ein-

fach auf den Ribbon-Eintrag **Entwurf**|**Externe Tabellendaten**|**Aktualisieren**. Kurz darauf erscheinen die Änderungen dann auch in der Excel-Tabelle.

Die Daten können so zwar in Excel angezeigt und dort auch bearbeitet werden, allerdings wirken sich Änderungen auf der Excel-Seite nicht auf die in der

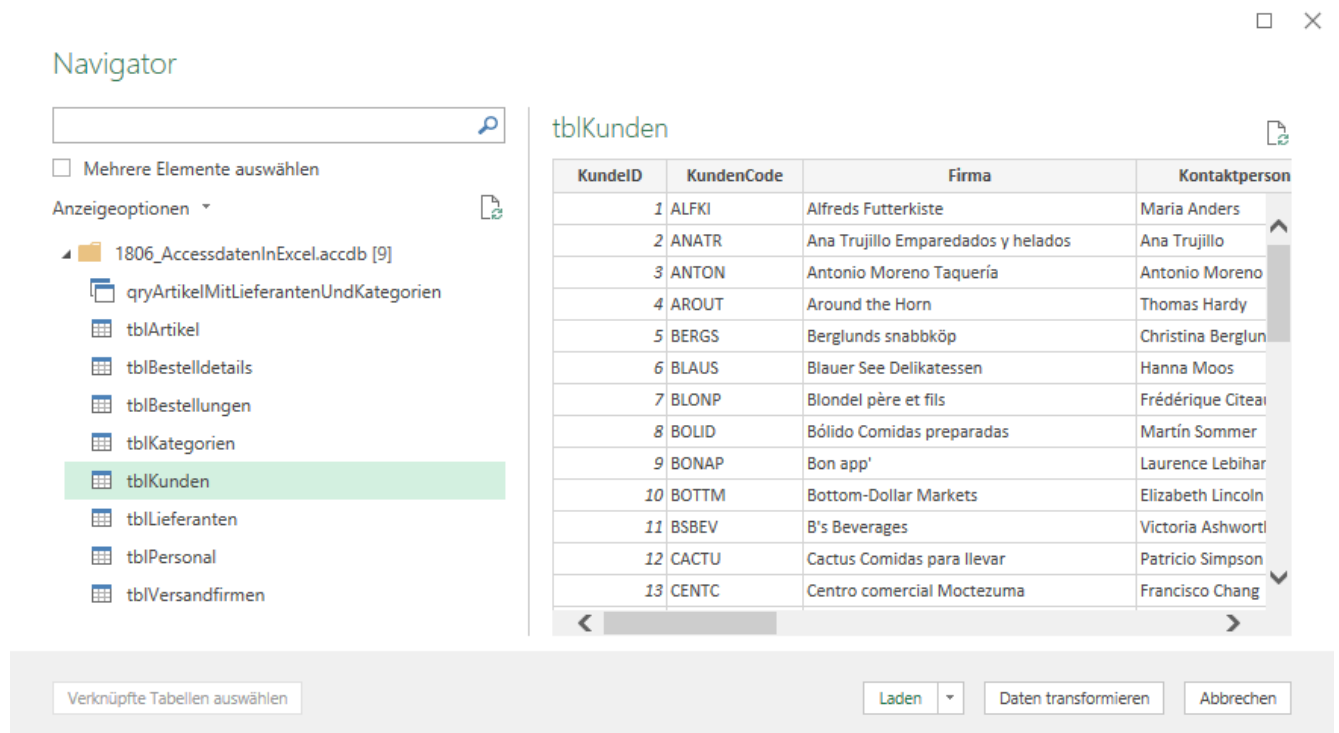


Bild 2: Auswahl der abzufragenden Daten

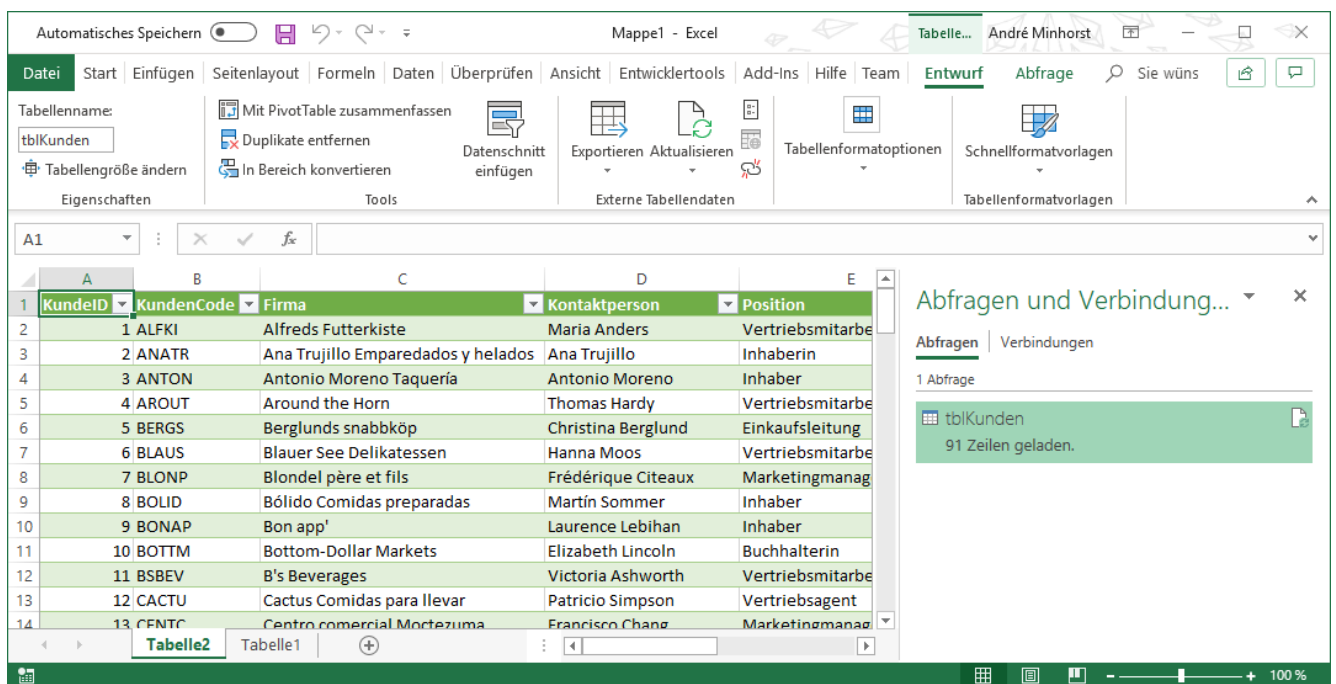


Bild 3: Die Daten der Access-Tabelle in Excel

Access-Datenbank gespeicherten Daten aus. Das ändert sich auch nicht, wenn Sie die Daten in Excel ändern und die Excel-Datei speichern – es bleibt eine Verknüpfung zu den Access-Daten, die von Excel aus nicht dauerhaft geändert werden kann.

Allerdings können Sie natürlich dennoch davon profitieren, dass die Daten nun in Excel vorliegen. Sie können damit beispielsweise Berechnungen durchführen oder Diagramme auf Basis dieser Daten erstellen.

Access-Verknüpfung in Excel von Access aus erstellen

Nun wollen Sie dem Benutzer vielleicht nicht die Arbeit aufbürden, Excel zu öffnen und von Hand die Verknüpfung zu den Daten der Access-Tabelle zu erstellen.

Deshalb zeigen wir Ihnen nun, wie Sie dies bequem von der Access-Anwendung aus erledigen können – per Mausklick auf eine Schaltfläche in einem Formular. Dieses soll direkt die in Excel anzuzeigenden Daten in einem Unterformular anzeigen.

Bevor wir das erledigen, fügen wir einen Verweis auf die Bibliothek **Microsoft Excel x.0 Object Library** hinzu.

Dazu öffnen Sie den VBA-Editor etwa mit der Tastenkombination **Alt + F11** und betätigen dann den Menübefehl **Extras|Verweise**. Hier suchen Sie nach dem gewünschten Eintrag und aktivieren diesen (siehe Bild 4).

Anschließend erstellen wir das Formular `frmKundenVerknuepfung`, dem wir das Unterformular `sfrmKunden` hinzufügen. Dieses enthält als Datensatzquelle die Tabelle `tblKunden` und zeigt die darin enthaltenen Daten in der Datenblattansicht an. Dem Hauptformular fügen wir zunächst eine Schaltfläche namens `cmdDatenInExcelVerknuepfen` hinzu (siehe Bild 5).

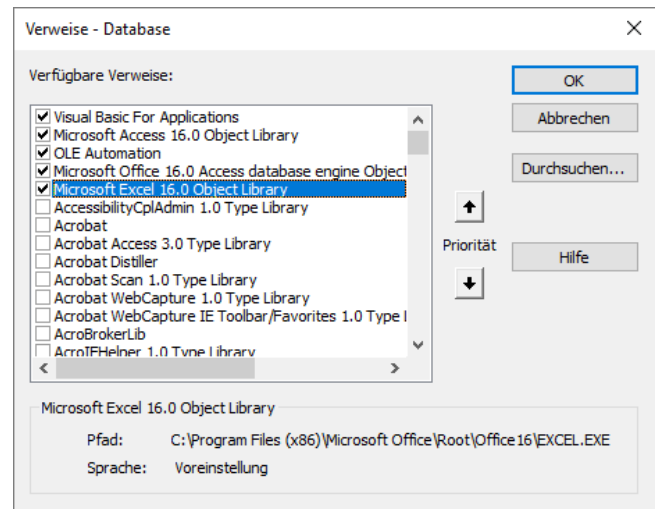


Bild 4: Verweis auf die Excel-Bibliothek

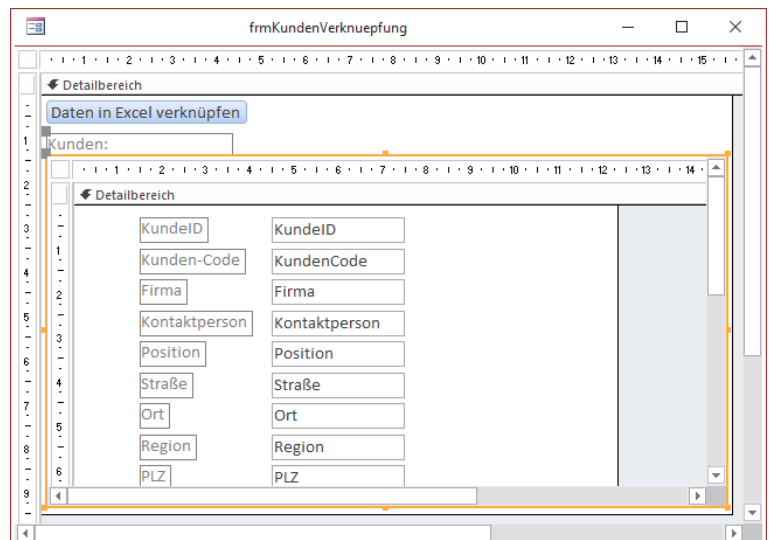


Bild 5: Entwurf des Formulars zum Öffnen der Verknüpfung in Excel

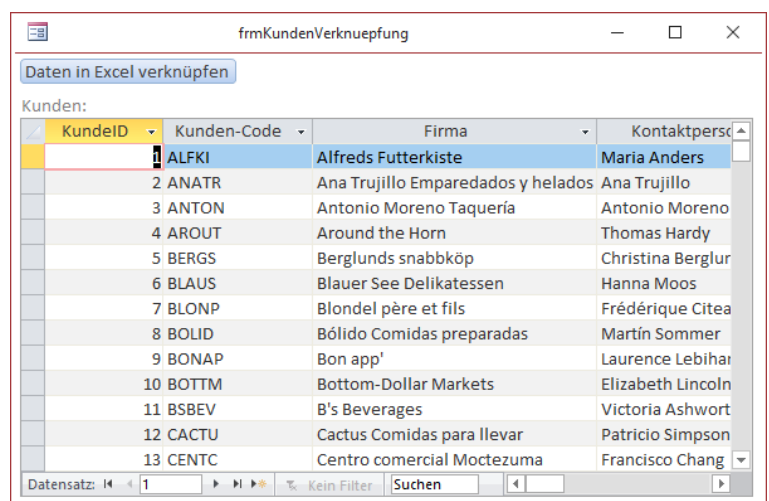


Bild 6: Formular mit den Daten, die in der Excel-Verknüpfung angezeigt werden sollen

In der Formularansicht sieht das Formular dann wie in Bild 6 aus.

Programmierung der Verknüpfung

Die Codezeilen zum Öffnen von Excel und zum Verknüpfen der Daten der Tabelle `tblKunden` fügen wir der Ereignisprozedur hinzu, die durch das Ereignis **Beim Klicken** der Schaltfläche `cmdDatenInExcelVerknuepfen` ausgelöst wird.

Zuvor deklarieren wir einige Variablen, die wir in mehreren Prozeduren benötigen:

```
Dim objExcel As Excel.Application
Dim objWorkbook As Excel.Workbook
Dim objQueryTable As Excel.QueryTable
```

`objExcel` referenziert dabei die Excel-Instanz, `objWorkbook` das darin geöffnete Workbook und `objQueryTable` die Repräsentation der Tabelle in Excel, welche die verknüpften Daten aus der Access-Datenbank (oder anderen Datenquellen) anzeigt.

Die Prozedur `cmdDatenInExcelVerknuepfen_Click` finden Sie in Listing 1.

Die Prozedur deklariert zunächst weitere Elemente: `objWorksheet` entspricht dem Tabellenblatt, in dem das `QueryTable`-Objekt angezeigt werden soll.

`objListobject` nimmt das Objekt auf, das direkt in das Tabellenblatt eingefügt wird und das `QueryTable`-Objekt enthält. `objRange` ist der Bereich, in dem das

```
Private Sub cmdDatenInExcelVerknuepfen_Click()
    Dim objWorksheet As Excel.Worksheet
    Dim objListobject As Excel.ListObject
    Dim objRange As Excel.Range
    Dim strConnection As String
    Dim strSQL As String
    Set objExcel = New Excel.Application
    objExcel.Visible = True
    Set objWorkbook = objExcel.Workbooks.Add
    On Error Resume Next
    Kill CurrentProject.Path & "\ExterneVerknuepfung.xlsx"
    On Error GoTo 0
    Set objWorksheet = objWorkbook.Sheets(1)
    Set objRange = objWorksheet.Range("A1")
    strSQL = "SELECT * FROM tblKunden"
    strConnection = "ODBC;DSN=MS Access Database;DBQ=" & CurrentDb.Name & ";DefaultDir=C:\path;DriverId=25;FIL=MS Access;7
                                                MaxBufferSize=2048;PageTimeout=5;"
    Set objListobject = objWorksheet.ListObjects.Add(xlSrcQuery, strConnection, , , objRange)
    Set objQueryTable = objListobject.QueryTable
    With objQueryTable
        .CommandText = strSQL
        .CommandType = xlCmdSql
        .RefreshOnFileOpen = True
        .Refresh False
    End With
    objWorkbook.SaveAs CurrentProject.Path & "\ExterneVerknuepfung.xlsx"
End Sub
```

Listing 1: Erstellen einer Excel-Datei mit einer Verknüpfung zu einer Access-Tabelle

ListObject-Objekt eingefügt wird. Und `strConnection` und `strSQL` nehmen die Verbindungszeichenfolge beziehungsweise die SQL-Anweisung auf.

Die erste richtige Anweisung erstellt eine neue Excel-Instanz und speichert diese in der Objektvariablen `objExcel`. Die zweite stellt die Eigenschaft `Visible` der Excel-Instanz auf `True` ein und macht die Instanz somit sichtbar. Dann fügt die Prozedur der Auflistung `Workbooks` mit der `Add`-Methode ein neues Element hinzu und referenziert dieses mit der Objektvariablen `objWorkbook`.

Danach löscht die Prozedur mit der `Kill`-Anweisung eine eventuell bereits vorhandene Datei namens `ExterneVerknuepfung.xlsx` aus dem Datenbankverzeichnis. Dabei wird die Fehlerbehandlung vorübergehend deaktiviert, damit kein Fehler ausgelöst wird, wenn die zu löschende Datei gar nicht vorhanden ist.

Anschließend referenziert die Prozedur das erste Tabellenblatt der Excel-Datei mit der Objektvariablen `objWorksheet` sowie die Zelle A1 als Bereich mit der Variablen `objRange`.

Die beiden folgenden Anweisungen weisen den String-Variablen `strSQL` und `strConnection` ihre Werte zu. Während `strSQL` nur eine einfache SQL-Anweisung aufnimmt, speichert `strConnection` die Verbindungszeichenfolge für den Zugriff von Excel auf die Access-Tabelle.

Dabei tauchen ein paar neue Parameter auf, die man vom Verknüpfen externer Quellen in Access nicht kennt. `DBQ` erwartet beispielsweise den Namen der Quelldatei, den übrigen Parametern werden Standardwerte zugewiesen.

Danach wird es interessant: Die folgende Anweisung fügt der `ListObjects`-Auflistung des `Worksheet`-

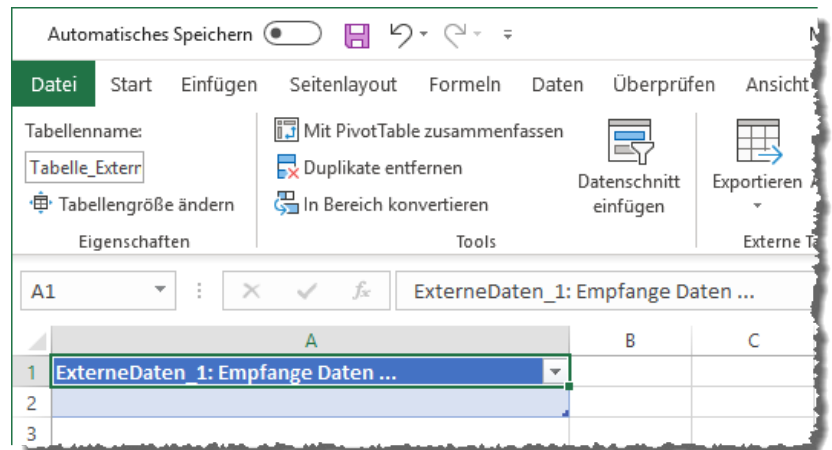


Bild 7: Erste Anzeige der kommenden verknüpften Daten

Objekts aus `objWorksheet` mit der `Add`-Methode ein neues Objekt hinzu. Dieser übergibt die Anweisung die Verbindungszeiche sowie den Range aus `objRange` als Parameter.

Das in `ListObjekt` enthaltene `QueryTable`-Objekt referenziert die Prozedur dann mit der Variablen `objQueryTable`. Dieses wiederum erlaubt die Angabe der SQL-Anweisung für den Zugriff auf die Daten der Access-Tabelle sowie einiger weiterer Parameter.

Schließlich speichert die Prozedur die frisch erstellte und mit der Verknüpfung ausgestattete Excel-Datei unter dem Namen `ExterneVerknuepfung.xlsx` im aktuellen Datenbankverzeichnis.

Direkt nach dem Ausführen dieser Prozedur und dem Erscheinen der Excel-Datei sieht das Arbeitsblatt wie in Bild 7 aus.

Wenige Sekunden später scheint dann auch die komplette Tabelle mit den enthaltenen Daten (siehe Bild 8).

Verknüpfte Daten von Access aus aktualisieren

Wir fügen dem Formular noch zwei weitere Schaltflächen zu (siehe Bild 9). Die erste soll die in Excel angezeigten Daten aktualisieren, sobald der Benutzer auf die Schaltfläche klickt. Das macht natürlich am

KundeID	KundenCode	Firma	Kontaktperson	Position
1	ALFKI	Alfreds Futterkiste	Maria Anders	Vertriebsmitarbeiter
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Inhaberin
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Inhaber
4	AROUT	Around the Horn	Thomas Hardy	Vertriebsmitarbeiter
5	BERGS	Berglunds snabbköp	Christina Berglund	Einkaufsleitung
6	BLAUS	Blauer See Delikatessen	Hanna Moos	Vertriebsmitarbeiter
7	BLONP	Blondel père et fils	Frédérique Citeaux	Marketingmanager
8	BOLID	Bóldo Comidas preparadas	Martín Sommer	Inhaber
9	BONAP	Bon app'	Laurence Lebihan	Inhaber
10	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Buchhalterin

Bild 8: Die Daten der verknüpften Tabelle in Excel

meisten Sinn, wenn auch die Daten in Access geändert wurden.

Die Aktualisierung der Daten wird durch die folgende Prozedur ausgelöst, welche die Refresh-Methode des QueryTable-Objekts aus objQueryTable aufruft:

```
Private Sub cmdVerknuepfung_Click()
    Aktualisieren_Click()
    With objQueryTable
        .Refresh
    End With
End Sub
```

Diese Prozedur ist auch der Grund, warum wir die Variable objQueryTable nicht in der Prozedur cmdDateInExcelVerknuepfen_Click deklariert haben, sondern im allgemeinen Teil des Klassenmoduls des Formulars frmKundenVerknuepfung.

Auf diese Weise können wir diese Variable in der einen Prozedur füllen

ter SaveChanges. Dadurch wird die Datei vor dem Schließen gespeichert.

Außerdem ruft sie die Quit-Methode von Excel auf und schließt somit die Excel-Instanz. Der Vollständigkeit halber setzen wir beide Objektvariablen noch auf den Wert Nothing:

```
Private Sub cmdExcelBeenden_Click()
    objWorkbook.Close True
    Set objWorkbook = Nothing
    objExcel.Quit
    Set objExcel = Nothing
End Sub
```

und in der nächsten ihre Requery-Methode aufrufen.

Excel von Access aus schließen

Fehlt noch eine Prozedur, welche die Excel-Datei von Access aus wieder schließt. Diese wird durch die Schaltfläche cmdExcelBeenden ausgelöst.

Diese schließt zunächst das Workbook mit der Close-Methode und übergibt dabei den Wert True für den Parameter

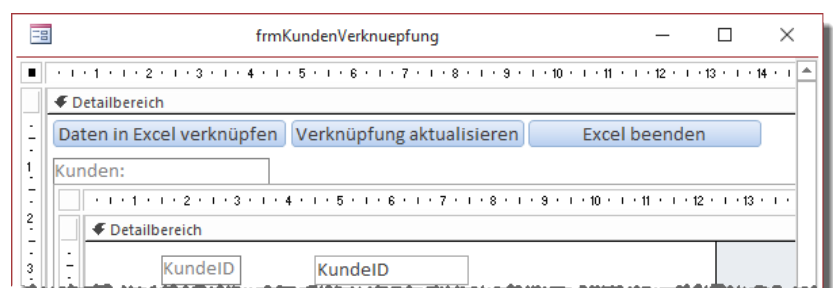


Bild 9: Schaltflächen zum Aktualisieren der Verknüpfung und zum Beenden von Excel