

MS Access 97

# CD-Verzeichnisse archivieren

Daten im MS-Access 97 speichern und abfragen

Karel Štípek

prg/cdarc.mdb

## Verwendung

Das Programm ermöglicht das Einlesen von Verzeichnissen mehrerer CDs in eine Tabelle. Die gespeicherten Daten können nach verschiedenen Kriterien gesucht und angezeigt werden. So kann schnell festgestellt werden, auf welcher CD sich eine bestimmte Datei befindet, ohne dass mehrere CDs durchgesucht werden müssen.

## Datenanalyse

Für jede Datei werden folgende Angaben gespeichert:

- Titel der CD
- Der komplette Pfad
- Name der Datei
- Erweiterung
- Eigendefinierte Notiz
- Alle oben genannten Attribute außer dem Pfad können auch als Suchkriterien auftreten.

Damit der CD-Titel nicht in jedem Datensatz gespeichert wird (unnötiger Speicherbedarf), werden die Titel in der Tabelle **tb1CD** gespeichert. Diese steht mit der Dateien-Tabelle **tb1Datei** in einer Beziehung 1:N, die Verknüpfung erfolgt durch den Schlüssel **1CD**.

Die gesamte Datenstruktur können Sie dem folgenden Bild entnehmen. Die Präfixe bei den Feldnamen bezeichnen die Datentypen. (Mehr über Datenanalyse und Namenskonventionen z.B. in **PCNEWS-53**, Seite 85 - „Kartei im MS-Access“).

## Programmwurf

Das Programm besteht aus einem einzigen Formular **frmHaupt**, das im unteren Teil das Unterformular **fsubDatei** für die Anzeige der detaillierten Datei-Informationen enthält.

Im oberen Teil rechts kann die logische Bezeichnung des CD Laufwerks eingegeben werden. Mit der Schaltfläche **CD einlesen** wird der Titel der CD abgefragt und das gesamte Verzeichnis in die Tabelle gefüllt.

Im mittleren Teil des Formulars können die Abfragekriterien eingegeben werden. Die CD kann in einem Kombobox ausgewählt werden, bei der Auswahl des Dateinamen, Erweiterung, bzw. Notiz können die Zeichen „?“ und „\*“ wie üblich verwenden

werden. Wenn in einem Auswahlfeld nichts eingegeben wird, wird das jeweilige Kriterium als „alle“ interpretiert.

Die Suche wird mit der Schaltfläche **Suchen** gestartet. Mit **Kriterien entfernen** werden alle Kriterien gelöscht und alle gespeicherten Datensätze angezeigt.

Im Unterformular können alle Datensätze geändert, gelöscht (oder auch neue eingegeben) werden. Zu jeder Datei kann auch eine eigene Notiz hinzugefügt werden.

## Realisierung

Der gesamte Programmcode ist im Klassenmodul des Hauptformulars gespeichert.

### CD einlesen

Die Ereignisprozedur der Schaltfläche **CD einlesen** ermöglicht zuerst die Eingabe des CD-Titels in einem standardmäßigen Dialogfenster und testet anschließend, ob das Laufwerk bereit ist. Wenn ein Fehler auftritt, kann nach Abfrage der Vorgang wiederholt werden.

```
Private Sub cmdEinlesen Click()
...
On Error GoTo err_einlesen
cdtitel =
  InputBox("Titel der eingelesenen CD",
    "Neue CD")
If cdtitel = "" Then Exit Sub
'Test, ob Laufwerk bereit
startpfad = Dir$(Me!txtlaufwerk & ".*")
...
err_einlesen:
If Err.Number = 76 Then
```

```
If MsgBox
  („Das CD-Laufwerk ist nicht bereit“, _
  vbRetryCancel) = vbRetry Then
  Resume
End If
...
```

Der CD-Titel wird in als ein neuer Datensatz in die Tabelle **tb1CD** geschrieben und der neue Wert des Schlüsselfelds wird in der privaten Variablen **cdkey** gespeichert. Dieser Schlüssel muss nämlich später in alle neu angelegten Datensätze der Tabelle **tb1Datei** geschrieben werden.

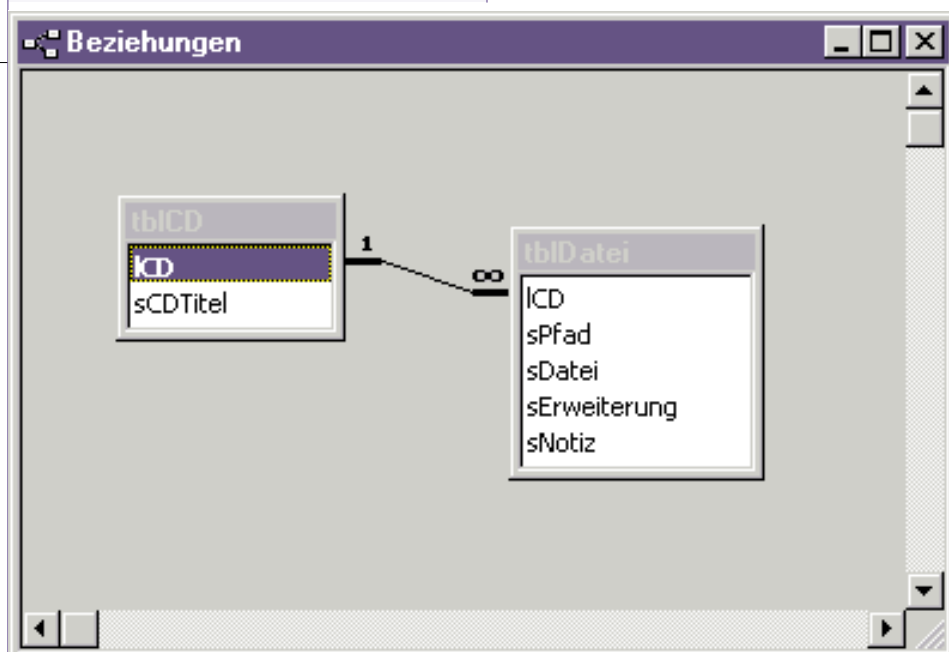
```
Set db = CurrentDb
Set rec = db.OpenRecordset(„tb1CD“)
rec.AddNew
rec!sCDTitel = cdtitel
rec.Update
rec.MoveLast
cdkey = rec!1CD
rec.Close
```

Das Einlesen der gesamten Verzeichnisstruktur ist ein rekursiver Vorgang, der in **PCNEWS-51**, Seite 102 - „Verzeichnisstruktur im EXCEL-VBA bearbeiten“ ausführlich beschrieben wurde.

Zuerst wird die Tabelle **tb1Datei** geöffnet und die Funktion **LesePfad()** mit dem Hauptverzeichnis als Parameter aufgerufen.

```
Set recDatei = db.OpenRecordset(„tb1Datei“)
startpfad = Me!txtlaufwerk & ".*"
LesePfad (startpfad)
```

Die Funktion **Dir\$()** bildet den Kern der Funktion **LesePfad()**. **Dir\$()** kann nicht rekursiv aufgerufen werden, deswegen müssen zuerst alle Einträge eines Unter-



The screenshot shows a window titled 'CD-Übersicht' with a search bar and a table of files. The search bar contains 'Monitor Sommer 1994'. The table has columns for 'CD', 'Pfad', 'Datei', 'Erw.', and 'Notiz'. The table lists various files from the 'Monitor Sommer 1994' CD, including system files like SESTART, SBYCR, SONY, SONY16, START, VCR, and application files like DEMOSASYMETRI, 3FOLIE, 4VORLAGE, STEKT, 6GRAFK, 70WAGRAM, 8VSGULL, 9MEDIA, BARCHIND, CPLBASE, and CPLM.

CD	Pfad	Datei	Erw.	Notiz
Monitor Sommer 1994	WVR	SESTART	BAT	
Monitor Sommer 1994	WVR	SBYCR	EXE	Demo
Monitor Sommer 1994	WVR	SONY	TXT	
Monitor Sommer 1994	WVR	SONY16	AVI	
Monitor Sommer 1994	WVR	START	BAT	
Monitor Sommer 1994	WVR	VCR	BMP	
Monitor Sommer 1994	WVR	VCR	EXE	Demo
Monitor Sommer 1994	DEMOSASYMETRI	2WERKZE	CPL	
Monitor Sommer 1994	DEMOSASYMETRI	3FOLIE	CPL	
Monitor Sommer 1994	DEMOSASYMETRI	4VORLAGE	CPL	
Monitor Sommer 1994	DEMOSASYMETRI	STEXT	CPL	
Monitor Sommer 1994	DEMOSASYMETRI	6GRAFK	CPL	
Monitor Sommer 1994	DEMOSASYMETRI	70WAGRAM	CPL	
Monitor Sommer 1994	DEMOSASYMETRI	8VSGULL	CPL	
Monitor Sommer 1994	DEMOSASYMETRI	9MEDIA	CPL	
Monitor Sommer 1994	DEMOSASYMETRI	BARCHIND	AAV	
Monitor Sommer 1994	DEMOSASYMETRI	CPLBASE	DLL	
Monitor Sommer 1994	DEMOSASYMETRI	CPLM	DLL	

verzeichnis im Array `datfeld` gespeichert werden.

Wenn das Element des Arrays `datfeld` ein Verzeichnis darstellt, wird die Funktion rekursiv aufgerufen, wenn es sich um eine Datei handelt, wird ein neuer Satz in der Tabelle `tblDatei` erstellt. Der Dateiname und die Erweiterung werden in getrennte Felder gespeichert.

```
Private Sub LesePfad(startpfad$)
.....
ReDim datfeld(0)
dateiname = Dir$(startpfad & "*.*",
vbDirectory)
```

```
Do While dateiname << " "
' Aktuelles und
' übergeordnetes Verzeichnis ignorieren.
If dateiname << "." And
dateiname << ".." Then
i = i + 1
ReDim Preserve datfeld(i)
datfeld(i) = dateiname
End If
dateiname = Dir
Loop
```

```
For i = 1 To UBound(datfeld)
dateiname = datfeld(i)
If (GetAttr(startpfad & dateiname) And
vbDirectory) = vbDirectory Then
LesePfad (startpfad & dateiname & "\")
Else
recDatei.AddNew
recDatei!ICD = cdkey
recDatei!sPfad = Mid$(startpfad, 3)
j = InStr(dateiname, ".")
If j > 0 Then
recDatei!sDatei =
Left$(dateiname, j - 1)
recDatei!sErweiterung =
Mid$(dateiname, j + 1)
Else
recDatei!sDatei = dateiname
End If
recDatei.Update
End If
Next i
End Sub
```

## Dateien suchen

Das Unterformular `fsubDatei`, wo alle gefundenen Datensätze angezeigt werden,

ist an die Abfrage `qryDiskDatei` gebunden. Diese Abfrage erfüllt zwei Aufgaben:

1. Zeigt zu jedem Datensatz der Tabelle `tblDatei` den zugehörigen CD-Titel an.
2. Selektiert die Daten je nach den eingegebenen Selektionskriterien.

Die Funktionalität einer Abfrage besteht eigentlich in der Interpretation eines SQL-Ausdrucks. SQL (*Structured Query Language*) ist eine Sprachdefinition, die in fast allen Datenbanksystemen implementiert ist.

Der SQL Ausdruck in unserer Abfrage enthält folgende Teile:

- Definition der selektierten Felder  
SELECT tblCD.ICD, tblCD.sCDTitle, tblDatei.sPfad, tblDatei.sDatei, tblDatei.sErweiterung, tblDatei.sNotiz
- Verbindung der Tabellen durch den CD-Schlüssel ICD

```
FROM tblCD LEFT JOIN tblDatei ON tblCD.ICD =
tblDatei.ICD
```

- Selektieren der Daten, die den eingegebenen Kriterien entsprechen. Ein leeres Formularfeld hat den Wert `Null`, die entsprechende Bedingung ist dann für alle Inhalte des Tabellenfelds erfüllt.

```
WHERE
((tblCD.ICD=forms!frmHaupt!cboCDTitle)
OR forms!frmHaupt!cboCDTitle is null)
AND ((tblDatei.sDatei like
forms!frmHaupt!txtDatei)
OR forms!frmHaupt!txtDatei is null)
AND ((tblDatei.sErweiterung like
forms!frmHaupt!txtErweiterung)
OR forms!frmHaupt!txtErweiterung is null)
AND ((tblDatei.sNotiz like
forms!frmHaupt!txtNotiz)
OR forms!frmHaupt!txtNotiz is null);
```

Nach dem Aktivieren der Schaltfläche **Suchen** wird das Unterformular aktualisiert.

```
Private Sub cmdSuchen Click()
Me!subDatei.Requery
End Sub
```

Die Ereignis-Prozedur der Schaltfläche **Kriterien entfernen** löscht zuerst die Inhalt

te aller Abfragefelder und aktualisiert dann das Unterformular.

```
Private Sub cmdInit Click()
Me!cboCDTitle = Null
Me!txtDatei = Null
Me!txtErweiterung = Null
Me!txtNotiz = Null
Me!subDatei.Requery
End Sub
```

## Weiterführende Literatur

Die Technik der Arbeit mit den Daten im MS-Access und eine Übersicht des *Visual Basics for Applications* findet man im unten abgebildeten Buch.

Das Buch beschäftigt sich nicht nur mit MS-Access, sondern geht auch komplexe Probleme an. Alle im Office97-Paket integrierten Anwendungen (Access, Word, Excel, PowerPoint, Outlook) werden aus der Sicht des gegenseitigen Datenaustausches behandelt. Sehr brauchbar finde ich auch die Kapitel über die Windows 95 - Registrierung, Win32API-Funktionen und Entwicklung eigener Hilfe.

*Integrierte Lösungen mit Office 97 von Gerhard Brosius, Christian Benkwitz, Norbert Böker und Tatjana Tegel. Verlag: Addison-Wesley-Longman 1997, ISBN 3-8273-1188-8. Dem Buch ist eine Diskette mit Beispielen beigelegt.*

