Familienstammbaum im Access

Karel Štípek

Die Darstellung eines Familienstammbaums ist keine ganz triviale Aufgabe. Besonder wichtig ist, beim Wechsel zwischen einzelnen Personen, bzw. Generationen die Übersicht nicht zu verlieren.

Im vorliegenden Artikel wird eine Lösung im Microsoft-Access präsentiert. Sie stellt ein praktisches Beispiel des relationalen Datenmodells und der Verwendung von SQL-Ausdrücken in den Formularsteuerelementen dar.

Die ganze Familie in einem Formular

Die Grundidee des vorgestellten Programms besteht darin, die ganze Verwandtschaft einer Person über drei Generationen (Eltern, Geschwister, Partner und Kinder) gleichzeitig in ein Formular hineinzubringen. Jede angezeigte Person kann (meistens mit Doppelklick) zur zentralen Person ausgewählt werden. Die Anzeige und Bezeichnung der Verwandten ändert sich dann dementsprechend automatisch. Wenn Sie zum Beispiel auf ein Kind doppelklicken, erscheint die vorher angezeigte Person als Mutter oder Vater und das Kind steht in der Mitte des Formulars.

Außer der Grunddaten (Name, Geburts-, bzw. Sterbedatum) und der Beziehungen können zu jeder Person zusätzliche Angaben gespeichert werden. Erstens sind es Informationen über die Wohnorte, zweitens Lebensereignisse oder beliebige andere Notizen. Sie können mit einem Datumsintervall versehen werden und können einen beliebig langen Text enthalten (Memo-Feld).

Bedienung des Programms

Benutzeroberfläche

Nachdem Sie die Datenbank **`Familie.mdb`** mit Access (ab Version 97) öffnen, erscheint das Hauptformular. Es besteht aus mehreren Unterformularen. In ihnen werden die Personen mit einer bestimmten Beziehung zur zentralen Person und die Wohnorte und Ereignisse dargestellt.

Alle Steuerelemente des Formulars sind mit den Quickinfo-Texten versehen. Damit wird die Bedienung erleichtert.

Beachten Sie, dass einige Teile des Formulars farbig sind. Durch die blaue, bzw. rosarote Farbe wird das Geschlecht der Personen hervorgehoben.

Testdaten

Die zur Verfügung gestellten Testdaten stellen keine konkrete Familie dar. Für die bessere Anschaulichkeit enthalten die Nachnamen direkt die Familienbeziehungen zu der Testperson. Die Datums- und Ortsangaben sind leer, weil sie für die Präsentation der Funktionalität des Programms irrelevant sind.

Zentrale Person

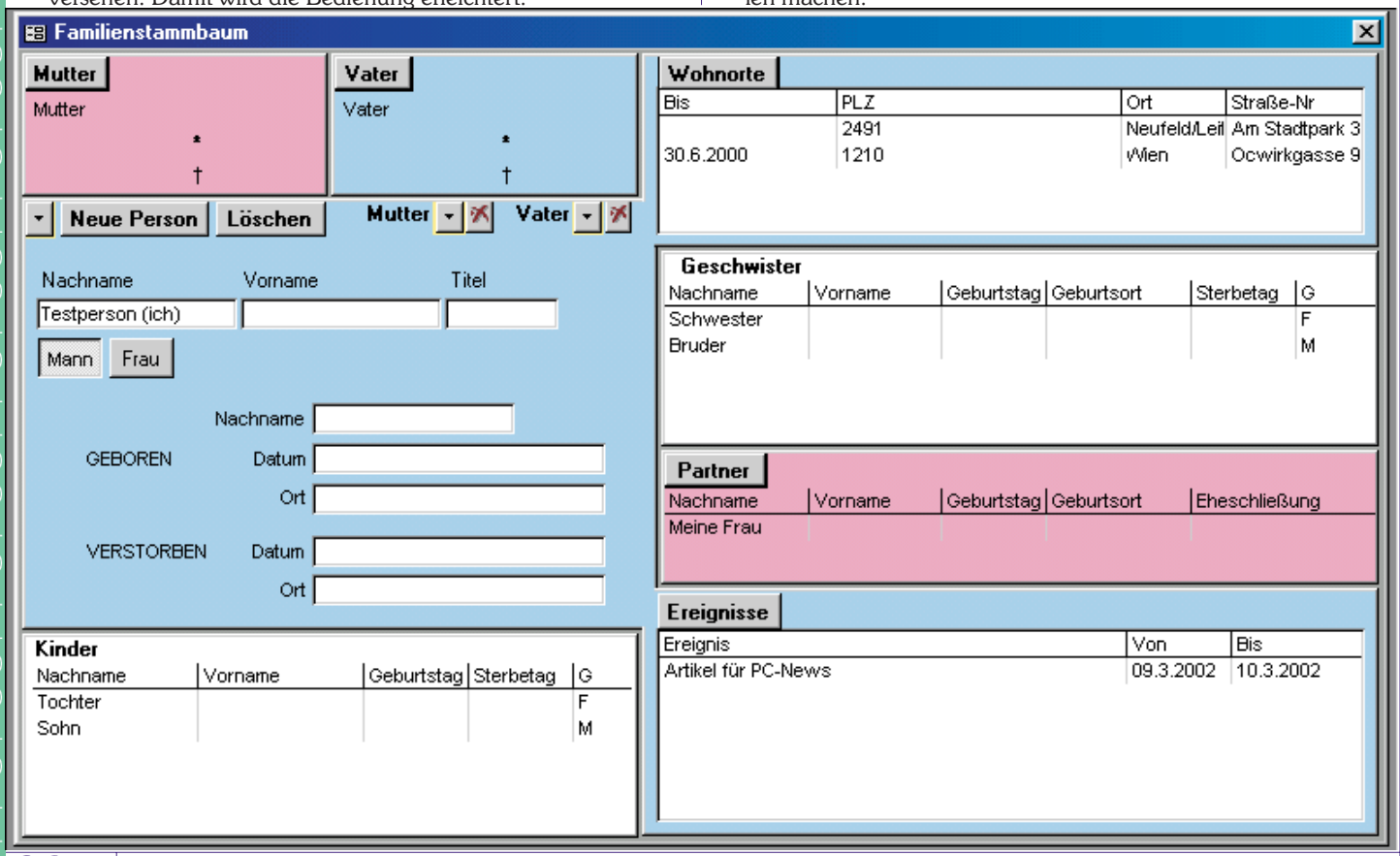
Als zentrale Person wird diejenige Person bezeichnet, die in der Mitte der linken Hälfte des Formulars angezeigt wird. Auf diese Person beziehen sich die Bezeichnungen der anderen Beziehungen und die Informationen über Wohnorte und Ereignisse. Alle angezeigten Felder außer Geschlecht (wird nur beim Anlegen einer neuen Person eingegeben) sind editierbar. Die Person kann mit der entsprechenden Schaltfläche nach einer Abfrage gelöscht werden.

Nach dem Programmstart wird die erste Person, die in der Personentabelle gespeichert wurde, angezeigt.

Auswahl der zentralen Person

Es gibt zwei Wege, wie Sie eine andere Person als zentrale Person anzeigen können:

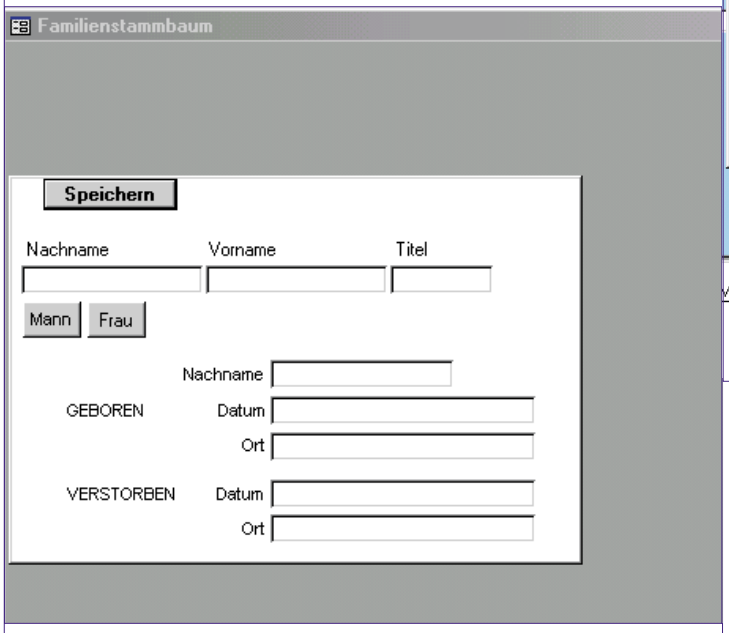
- Mit Hilfe der Kombobox in der linken oberen Ecke (nur das Rechteck mit Pfeil ist sichtbar) können Sie eine beliebige Person direkt aus allen bereits gespeicherten Personen auswählen. In der Auswahlliste werden Nachname, Vorname, Titel und Geburtsdatum angezeigt.
- Mit dem Doppelklick auf eine Person aus Geschwistern, Partnern oder Kindern, bzw. mit dem einfachen Klick auf die Schaltflächen beim Vater oder Mutter können Sie die jeweilige Person zur zentralen machen.



Nachdem die zentrale Person geändert wurde, wird die Anzeige aller anderen Familienmitglieder sofort dementsprechend geändert.

Neue Person anlegen

Nachdem Sie die Schalfläche ***Neue Person*** anklicken, werden alle Teile des Formulars (bis auf den zentralen) ausgeblendet und Sie können die Daten einer neuen Person eintragen. Es muss mindestens der Nachname und das Geschlecht eingegeben werden. Nach dem Speichern bleibt die zuletzt gespeicherte Person als zentrale Person angezeigt. Die restlichen Teile des Formulars bleiben leer, weil noch keine Beziehung und keine zusätzlichen Informationen (Wohnorte, Ereignisse) eingegeben sind.



Eingliederung in die Familie

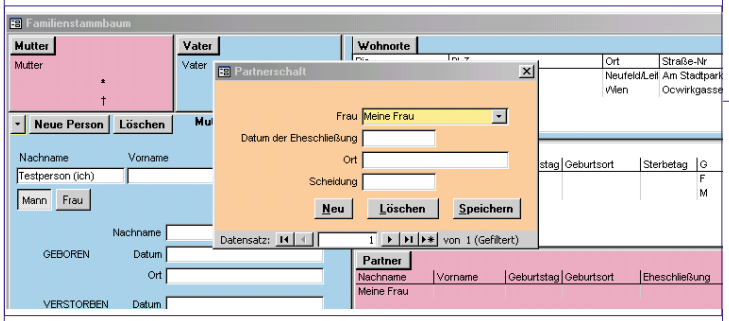
Jede Person bekommt ihren richtigen Platz in der Familienhierarchie, indem ihr Vater und Mutter zugeordnet werden. Es wird davon ausgegangen, dass die Eltern als Personen mit dem oben erklärten Verfahren bereits angelegt wurden.

In der rechten oberen Ecke des zentralen Teils befinden sich zwei Komboboxen (wieder ohne das Eingabefeld) für die Zuordnung der Mutter und des Vaters. Es werden natürlich nur Personen mit dem richtigen Geschlecht zur Auswahl angeboten. Nachdem Sie beide Elternteile definiert haben, können Sie sofort alle anderen bereits verfügbaren Verwandten sehen.

Die Zuordnung der Mutter oder des Vaters können Sie entweder jederzeit ändern oder mit den kleinen Schaltflächen neben den Komboboxen entfernen.

Partnerschaft definieren

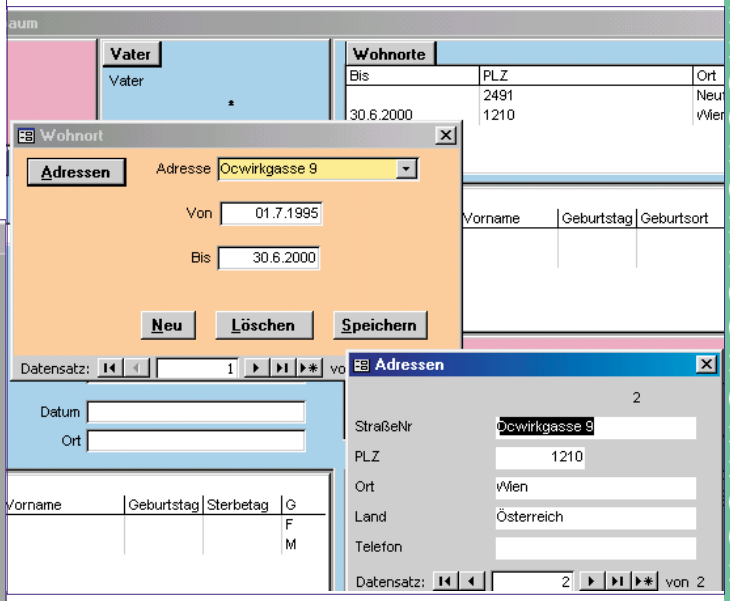
Nachdem Sie in der Mitte der rechten Formularhälfte auf die Schaltfläche ***Partner*** klicken, wird das Formular ***Partnerschaft*** geöffnet. Hier können Sie der zentralen Person einen oder mehrere Partner zuordnen und das Datum und Ort der Eheschließung, bzw. das Scheidungsdatum eintragen.



Es wird vorausgesetzt, dass der Partner/Partnerin vorher als eine Person gespeichert wurde.

Wohnorte

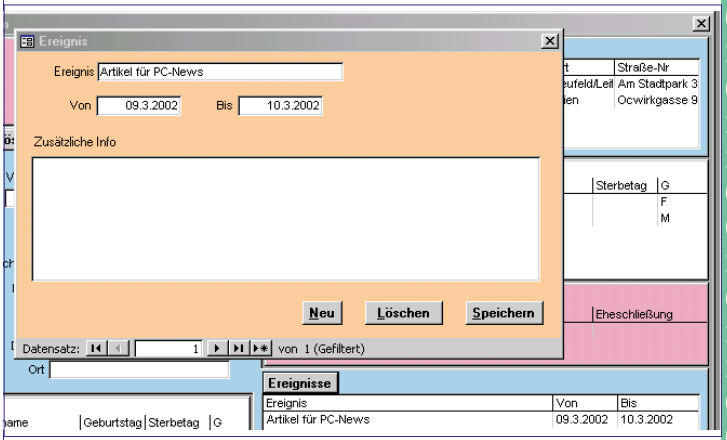
Damit die Adressen nicht mehrmals für mehrere zusammenlebende Personen eingegeben werden müssen, werden sie getrennt von der Zuordnung zur Person gespeichert. Die Struktur ist ersichtlich, wenn Sie mit der Schalfläche ***Wohnorte*** das Formular ***Wohnort*** öffnen. In der Kombobox oben können Sie die bereits gespeicherten Adressen auswählen und das Von- , bzw. Bis-Datum eintragen.



Die Verwaltung der Adressen wird in einem getrennten Formular durchgeführt, das Sie mit der Schaltfläche ***Adressen*** öffnen können. Die in diesem Formular eingegebenen Daten sind von den Personenangaben unabhängig. Erst in dem Formular ***Wohnort*** werden sie der jeweiligen Person zugeordnet.

Ereignisse

Das Formular ***Ereignis*** ist für das Speichern von beliebigen Lebensereignissen oder anderen Notizen bestimmt. Auf dem Hauptformular werden das Titel und das Von- , bzw. Bis-Datum angezeigt. Außerdem können Sie zu jedem Eintrag noch einen beliebig langen Text (Memofeld) speichern.



Sowohl die Wohnorte als auch die Ereignisse werden immer der Person zugeordnet, die bereits als die zentrale Person angezeigt wird.

Datenanalyse der Familienstruktur

Personenschlüssel

Für die richtige und zuverlässige Definition der Familienbeziehungen ist es notwendig, jede Person mittels eines eindeutigen Schlüssels zu identifizieren. Aus diesem Grund wurde in der Tabelle `tblPerson` ein Autowert-Feld `lPerson` als primärer Schlüssel erstellt, das die automatische Nummerierung der Personen garantiert.

Der Präfix "`l`" steht für Datentyp `Long Integer` und wird für alle Felder verwendet, die den Verweis auf eine Person darstellen. Darüber erfahren Sie mehr bei der Beschreibung der einzelnen Tabellen der Applikation.

Tabelle tblPerson

In der Tabelle tblPerson werden alle Angaben zu einer Person und die Zuordnung zu Mutter und Vater gespeichert. Sie enthält folgende Felder:

Feldname	Typ	Bedeutung
lPerson	Autowert	Primärer Schlüssel
lMutter	LongInt	Schlüssel der Mutter
lVater	LongInt	Schlüssel des Vaters
sNachname	Text	
sVorname	Text	
sTitel	Text	
sGebNachname	Text	GeburtsNachname
iGeschlecht	Integer	1=Mann, 2=Frau
dtmGeburtstag	Datum	
sGeburtsort	Text	
dtmSterbetag	Datum	
sSterbeort	Text	
sLand	Text	
sTelefon	Text	

Mit der bereits vorgestellten Personentabelle kann eigentlich die Familienhierarchie mit Eltern und Kindern einer Person abgebildet werden, wenn bei jeder beteiligten Person die Felder lMutter und lVater mit den richtigen Personenschlüsseln befüllt werden.

Tabelle tblPartner

Für die Abbildung einer Partnerschaft brauchen wir eine weitere Tabelle, welche in einem Datensatz zwei zusammengehörende Personenschlüssel speichern kann. Diese Tabelle heißt tblPartner und hat folgende Struktur:

Feldname	Typ	Bedeutung
lMann	LongInt	Schlüssel des Mannes
lFrau	LongInt	Schlüssel der Frau
dtmVon	Datum	Datum der Eheschließung
sOrtVon	Text	Ort der Eheschließung
dtmBis	Datum	Datum der Scheidung

Im Sinne des relationellen Datenmodells handelt es sich in diesem Fall um eine Tabelle, die die Tabellenbeziehung M:N vermittelt (sowohl Männer als auch Frauen können Beziehungen mit mehreren Personen bilden).

Tabelle tblAdresse

Die Adressentabelle erfordert keine besondere Erklärung. Sie enthält ein Autowert-Feld lAdresse, welches den primären Schlüssel darstellt. Sein Wert tritt in der Wohnort-Tabelle als fremder Schlüssel auf, um auf die Adresse zu verweisen.

Feldname	Typ	Bedeutung
lAdresse	AutoWert	Primärer Schlüssel
sStraßeNr	Text	
lPLZ	LongInt	
sOrt	Text	

Tabelle tblWohnort

Die Tabelle tblWohnort hat eine ähnliche Funktion wie die Tabelle tblPartner. Sie speichert die primären Schlüssel der Tabellen tblPerson und tblAdresse, die in einer m:n-Beziehung stehen. In der Tabelle wird zusätzlich noch die Dauer dieser Beziehung (Gültigkeit des jeweiligen Wohnortes) abgespeichert.

Feldname	Typ	Bedeutung
lPerson	LongInt	Schlüssel der Person
lAdresse	LongInt	Schlüssel der Adresse
dtmVon	Datum	
dtmBis	Datum	

Tabelle tblEreignis

Die Ereignistabelle kann mehrere Lebensereignisse (oder beliebige andere Informationen) zu einer Person speichern. Es wird nicht vorausgesetzt, dass wie bei Adressen mehrere Personen die gleichen Ereignisse teilen. Aus diesem Grund ist hier eine 1:n-Beziehung ohne eine zusätzliche Tabelle (wie tblPartner oder tblWohnort) realisiert.

Die Struktur ist einfach und übersichtlich. Der einzige Schlüsselwert ist der fremde Schlüssel lPerson, der die Verbindung zu einer Person darstellt.

Feldname	Typ	Bedeutung
lPerson	LongInt	Schlüssel der Person
sEreignis	Text	Titel
dtmVon	Datum	
dtmBis	Datum	
memText	Memo	Zusatzinfo

Darstellung der Familienbeziehungen

Zusammenhänge zwischen den Schlüsselwerten

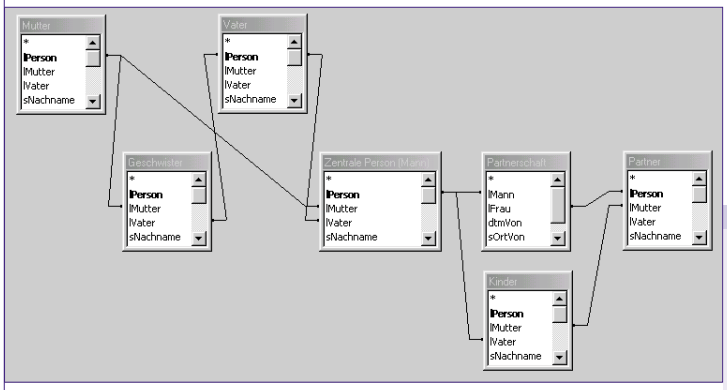
Ausgehend aus den bereits vorgestellten Tabellenstrukturen können die grundlegenden Zusammenhänge zwischen den Schlüsselwerten abgeleitet werden:

- Als Vater, bzw. Mutter wird diejenige Person angezeigt, derer Schlüssel im Feld lVater oder lMutter der zentralen Person steht.
- Prinzipiell die gleiche Regel gilt für die Kinder: es sind die Personen, die im Feld lVater oder lMutter den Schlüssel der zentralen Person enthalten.
- Geschwister sind diejenige Personen, die mindestens in einem der Felder lVater oder lMutter den gleichen Wert wie die zentrale Person enthalten.
- Partner werden mittels der Tabelle tblPartner identifiziert. Es werden alle Personen aufgelistet, derer Schlüssel in gleichen Datensätzen wie der Schlüssel der zentralen Person steht.
- Anzeige der Wohnorte und Ereignisse ist einfach. Die Datensätze, die den Schlüssel der zentralen Person enthalten, werden aufgelistet.

Graphische Darstellung der Familienhierarchie

Auf dem folgenden Bild können Sie die bereits angeführten Zusammenhänge anschaulich sehen. Das Bild ist als Entwurfsicht einer Abfrage erstellt worden. Diese Abfrage steht zwar in der gelieferten Datenbank zur Verfügung, ist aber nicht für eine Datenanzeige bestimmt und wird auf im Programm als Abfrage nicht eingesetzt.

Alle dargestellten Tabellen außer der Partnerschaftstabelle sind als unterschiedliche Datesätze der gleichen Personentabelle zu verstehen.



Funktionsanalyse - allgemeine Vorgänge

SQL als Hauptwerkzeug

Die Funktionalität des Programms besteht grundsätzlich aus einer Menge von Datenselektionen und Anzeige der Ergebnisse in Unterformularen je nach der Beziehung zu der bereits angezeigten zentralen Person. Die für die Selektion notwendigen SQL-Ausdrücke sind meistens direkt als Datenherkunft des jeweiligen Formulars oder Steuerelements im Eigenschaftsfenster eingetragen. Nur in drei Fällen, wenn mehrere Tabellen verknüpft werden, wurden die Selektionen als benannte Abfragen definiert.

Struktur des Hauptformulars

Das Hauptformular `frmHaupt` ist an die Tabelle `tblPerson` gebunden, obwohl keine Datenfelder im Formular direkt angezeigt oder bearbeitet werden. Anzeige aller gespeicherten Daten erfolgt in Unterformularen, die den jeweiligen Teil der Familienhierarchie, bzw. die Zusatzangaben (Wohnorte und Ereignisse) samt der notwendigen Steuerelementen enthalten.

Referenz auf die zentrale Person

Der primäre Schlüssel der zentralen Person ist das wichtigste Selektionskriterium. Alle angezeigten Familienmitglieder, bzw. ihre Position in der Hierarchie werden im Bezug auf die zentrale Person selektiert.

In den SQL-Selektionsausdrücken wird folgende Referenz auf das Haupformularfeld, bzw. Datenfeld der Personentabelle verwendet:

```
Forms!frmHaupt!lPerson
```

Alle Formulare, in denen neue Datensätze im Bezug auf die zentrale Person angelegt werden, übernehmen in der Ereignis-Prozedur `BeforeInsert` den o.g. Wert des Personenschlüssels in den neu erstellten Datensatz. Es betrifft die Formulare `frmPartner`, `frmWohnort` und `frmEreignis`, die später noch diskutiert werden.

```
Sub Form_BeforeInsert(Cancel As Integer)
  Me!lPerson = Forms!frmHaupt!lPerson
End Sub
```

Auswahl der zentralen Person

Nachdem eine neue zentrale Person ausgewählt wird, wird die Funktion `SelectPerson()` aufgerufen. Ihre Aufgabe besteht darin, die Daten für das Hauptformular zu filtern. Die Funktion `FarbeNachGeschlecht()` ändert die Farbe einiger Unterformulare je nach dem aktuellen Geschlecht. Es ist aber eigentlich nur bei der zentralen Person, Mutter, Vater und Partner eindeutig möglich. Bei Kindern und Geschwistern können in einem Unterformular sowohl männliche als auch weibliche Personen auftreten.

Die Funktion `RequeryAll()` aktualisiert die Anzeige der Personen in allen Unterformularen und aktualisiert auch den Inhalt aller Komboboxen, in welchen Personen je nach dem Familienverhältnis zur zentralen Person ausgewählt werden können.

```
Function SelectPerson(person As Variant)
If Not IsNull(person) Then
  DoCmd.ApplyFilter , "lPerson = " & person
  FarbeNachGeschlecht
  RequeryAll
End If
```

Unterfomulare färben

Die Funktion `FarbeNachGeschlecht()` ändert die Hintergrundfarbe einiger Unterformulare je nach dem Geschlecht der zentralen Person. Die Farbenwerte sind im globalen Modul "`g`" als Konstanten `FARBE_MANN`, `FARBE_FRAU` und `FARBE_WEISS` gespeichert. Die weiße Farbe ist für die Eingabe einer neuen Person bestimmt (bevor das Geschlecht angegeben wird).

Die Funktion hat noch eine weitere Aufgabe. Je nach dem Geschlecht der zentralen Person wird die Datensatzherkunft für die Listbox, in dem die Partner angezeigt werden, geändert.

```
Function FarbeNachGeschlecht(geschlecht%)
If geschlecht = MANN Then
  farbe = FARBE_MANN
  partnerfarbe = FARBE_FRAU
  Forms!frmHaupt!subPartner.Form!lstPartner.
  RowSource = "qryPartnerFrau"
ElseIf geschlecht = FRAU Then
  farbe = FARBE_FRAU
  partnerfarbe = FARBE_MANN
  Forms!frmHaupt!subPartner.Form!lstPartner.
  RowSource = "qryPartnerMann"
Else
  farbe = FARBE_WEISS
  partnerfarbe = FARBE_WEISS
End If
Forms!frmHaupt!subPerson.Form.
Detailbereich.BackColor = farbe
...
```

Anzeige aktualisieren

Nach dem Wechsel oder Löschen der zentralen Person oder einer Änderung der Familienbeziehungen muss die Anzeige in allen Unterformularen aktualisiert werden. Diese Aufgabe übernimmt die Funktion `RequeryAll()`. Sie besteht nur aus einer Reihe von Aufrufen der Methode `Requery` für alle betroffenen Steuerelemente.

```
Function RequeryAll()
  Set frm = Forms!frmHaupt
  frm!subKinder.Form!lstKinder.Requery
  ...
```

Eingabe einer neuen Person

Für die Eingabe der Daten einer neuen Person wird der zentrale Teil des Hauptformulars verwendet. Alle restlichen Teile werden mit der Funktion `RelatVisible()` ausgeblendet. Für den Start der Eingabe und Speichern des neuen Datensatzes wird die gleiche Schaltfläche eingesetzt. Ihre Aufschrift wird von "***Neue Person***" auf "***Speichern***" temporär geändert.

In der Ereignisprozedur `cmdNeuePerson_Click()` wird die Aufschrift auf der Schaltfläche abgefragt und entweder die Eingabe gestartet oder der neue Datensatz gespeichert.

Nach dem Speichern einer neuen Person wird sie als zentrale Person angezeigt. Sollte die Eingabe unterbrochen werden, wird die Anzeige der vorherigen Person wiederhergestellt. Ihr Schlüssel wird in der statischen Variablen `savperson` nach dem Start der Eingabe abgelegt.

```
Sub cmdNeuePerson_Click()
Static savperson&
  If Me!cmdNeuePerson.Caption = "Neue Person"
  Then
    If Not IsNull(Me!lPerson) Then
      savperson = Me!lPerson
    End If
    FarbeNachGeschlecht (0)
    Me!cmdNeuePerson.Caption = "Speichern"
    RelatVisible (False)
    ...
    DoCmd.RunCommand (acCmdDataEntry)
  Else
    If IsNull(Me!txtNachname) Or
    (Me!grpGeschlecht = 0) Then
      If MsgBox("Nachname und Geschlecht
          müssen eingegeben werden",
          vbRetryCancel) = vbRetry Then
        Exit Sub
      Else
        sav = False
      End If
    Else
      sav = True
    End If
    Me!cmdNeuePerson.Caption = "Neue Person"
    RelatVisible (True)
```

```
    ...
    If sav Then
      DoCmd.RunCommand (acCmdSaveRecord)
      savperson = DLast("lPerson",
                        "tblPerson")
    End If
    If Not IsNull(savperson) Then
      SelectPerson (savperson)
    End If
  End If
```

Funktion RelatVisible()

Die Funktion RelatVisible() blendet die Unterformulare für Verwandte, Wohnnorte und Ereignisse und einige Steuerelemente des zentralen Unterformulars je nach dem Wert des Parameters visib ein oder aus. Sie wird ausschließlich bei der Eingabe einer neuen Person eingesetzt.

```
Function RelatVisible(visib%)
  Set frm = Forms!frmHaupt
  Set subpers = frm!subPerson.Form

  frm!subMutter.Visible = visib
  ...
  subpers!cboPerson.Visible = visib
  ...
```

Funktionsanalyse - Datenselektionen

Auswahl einer beliebigen Person

Der einfachste Selektionsausdruck wird auf dem Unterformular fsubPerson in der Kombobox cboPerson als die Eigenschaft Datensatzherkunft eingetragen. Er liefert die Felder, die bei der Auswahl einer Person angezeigt werden.

```
SELECT lPerson,sNachname, sVorname, sTitel, dtmGeburtstag
FROM tblPerson
ORDER BY sNachname, dtmGeburtstag
```

Zuordnung der Mutter und des Vaters

Die SQL-Ausdrücke in der Eigenschaft Datensatzherkunft der Kombobox für die Auswahl der Mutter und des Vaters sind beide identisch bis auf das Geschlecht.

Es werden aber nur die Personen selektiert, die älter als die zentrale Person sind und zum Zeitpunkt ihrer Geburt nicht tot waren.

```
SELECT ... FROM tblPerson
WHERE (iGeschlecht=2)
AND (dtmGeburtstag<
Forms!frmHaupt!subPerson.Form!dtmGeburtstag) ...
AND (dtmSterbetag>=
Forms!frmHaupt!subPerson.Form!dtmGeburtstag)
...
```

Die Komboboxen cboMutter und cboVater sind direkt an die Felder lMutter, bzw. lVater gebunden. Nach der Auswahl der Person wird ihr Schlüssel direkt in den Datensatz der zentralen Person gespeichert. In der Ereignis-Prozedur Nach Aktualisierung wird die Funktion SelectPerson() aufgerufen. Damit werden die bei der Zuordnung schon verfügbaren Verwandten sofort angezeigt.

Anzeige der Mutter und des Vaters

Die SQL-Ausdrücke für die Datenherkunft des Unterformulars subMutter entsprechen genau der o.g. Definition.

```
SELECT * FROM tblPerson
WHERE
(tblPerson.lPerson = Forms!frmHaupt!lMutter)
```

Der analogische SQL-Ausdruck für das Unterformular subVater lautet:

```
SELECT * FROM tblPerson
WHERE
(tblPerson.lPerson = Forms!frmHaupt!lVater)
```

Anzeige der Kinder

Die Kinder werden im Unterformular fsubKinder, in der Listbox lstKinder angezeigt. Das nummerische Feld Geschlecht wird vor der Anzeige der Kinder oder Geschwister in ein "M" oder "F" umgewandelt und bekommt einen kurzen Namen "G". Die Eigenschaft Datensatzherkunft dieser Listbox enthält folgenden SQL-Ausdruck:

```
SELECT ..., iif(iGeschlecht=1,"M","F") AS G
FROM tblPerson
WHERE
tblPerson.lMutter=Forms!frmHaupt!lPerson
OR
tblPerson.lVater=Forms!frmHaupt!lPerson
ORDER BY dtmGeburtstag;
```

Anzeige der Geschwister

Die Selektion der Geschwister ist ähnlich wie bei den Kindern. Das Geschlecht wird wieder explizit angezeigt, weil es von dem Geschlecht der zentralen Person nicht eindeutig abgeleitet werden kann.

Die oben angeführte Definition der Geschwister muss noch etwas präzisiert werden.

- Aus der Übereinstimmung der Schlüsselwerte lMutter oder lVater muss der Wert 0 ausgeschlossen werden, sonst werden alle Personen ohne zugeordnete Eltern als Geschwister angezeigt.
- Die zentrale Person darf nicht gleichzeitig unter den Geschwistern erscheinen.

Zwischen den zwei Bedingungen für die gleiche Mutter oder den gleichen Vater steht der logische Operator OR, damit auch Halbgeschwister ausgewählt werden.

```
SELECT ..., iif(iGeschlecht=1,"M","F") AS G
FROM tblPerson
WHERE
(((tblPerson.lMutter=Forms!frmHaupt!lMutter) AND
(tblPerson.lMutter>0))
OR
((tblPerson.lVater=Forms!frmHaupt!lVater)
AND (tblPerson.lVater>0)))
AND
(tblPerson.lPerson<>Forms!frmHaupt!lPerson)
ORDER BY dtmGeburtstag;
```

Zuordnung der Partner

Mit der Schaltfläche ***Partner*** wird das Formular frmPartner geöffnet. Dieses Formular ist an die Tabelle tblPartner gebunden. Im oberen Teil sind zwei Komboboxen cboMann und cboFrau für die Auswahl des Mannes, bzw. der Frau.

Die SQL-Ausdrücke für diese Komboboxen unterscheiden sich nur in der Auswahl des Geschlechts. Außerdem werden die Personen ausgeschlossen, die bei der Geburt der zentralen Person bereits tot waren. Die Datensatzherkunft z.B. für die Kombobox cboMann lautet also:

```
SELECT ...  FROM tblPerson
WHERE (iGeschlecht=1)
AND
((dtmSterbetag>=
Forms!frmHaupt!subPerson.Form!dtmGeburtstag) OR
IsNull(tblPerson.dtmSterbetag)
OR IsNull
(Forms!frmHaupt!subPerson.Form!dtmGeburtstag))
ORDER BY sNachname, dtmGeburtstag
```

Beim Laden des Formulars wird je nach dem Geschlecht der zentralen Person nur eine dieser Komboboxen angezeigt. Außerdem werden nur die Datensätze der Partnerschafttabelle ausgefiltert, die im Feld lMann, bzw. lFrau den Schlüssel der zentralen Person enthalten.

```
Sub Form_Load()
  If Forms!frmHaupt!iGeschlecht = 1 Then
    Me!cboMann.Visible = False
    Me!cboFrau.Visible = True
    DoCmd.ApplyFilter ,
          "lMann = Forms!frmhaupt!lPerson"
  Else
    Me!cboMann.Visible = True
    Me!cboFrau.Visible = False
    DoCmd.ApplyFilter ,
          "lFrau = Forms!frmhaupt!lPerson"
  End If
```

Die Komboboxen sind an die Felder lMann, bzw. lFrau gebunden, dadurch wird nach der Auswahl der Personenschlüssel der ausgewählten Person in die Partnerschaftstabelle gespeichert.

Anzeige der Partner

Die Listbox lstPartner auf dem Unterformular fsubPartner wird in der Funktion FarbeNachGeschlecht() je nach dem Geschlecht an die Abfrage qryPartnerMann oder qryPartnerFrau gebunden.

Die o.g. Abfragen müssen auf zwei Tabellen zugreifen. Es werden die Personen aufgelistet, deren Schlüssel in der Tabelle tblPartner gemeinsam mit dem Schlüssel der zentralen Person steht.

Java

Textdateien durchsuchen

Peter Nussbaumer, Alfred Nussbaumer

Java ist grundsätzlich für grafisch orientierte Anwendungen entwickelt worden; mit dem *Abstract Window Toolkit (awt)* bzw. mit den Swingkomponenten werden Dateizugriffe relativ einfach realisiert. Im Unterricht spielen jedoch kurze, textbasierte Beispiele eine wichtige Rolle. In diesem Beitrag soll die textbasierte Ein- und Ausgabe dargestellt werden.

Java-Applets können aus Sicherheitsgründen keine Dateien lesen oder schreiben. Diese Beschränkungen gelten für Applikationen nicht, sondern die Lese- und Schreibrechte entsprechen den Dateizugriffsrechten des jeweiligen Benutzers.

In den folgenden Beispielen sollen zwei Applikationen vorgestellt werden, die zeilenweise bzw. byteweise Zeichen aus Dateien auslesen, die die derzeit größte bekannte Primzahl enthalten. Die Datei **„prime5.txt"** enthält die Ziffern der Primzahl in Zeilen zu je 80 Zeichen (sie steht unter http://www.mersenne.org zum Download bereit); die Datei **„prime.txt"** enthält alle Ziffern in einer einzigen Bytefolge. Beim Lesen aus einer Datei werfen die verwendeten Methoden so genannte *„Exceptions"*, die eine geordnete Fehlerbehandlung erlauben (man beachte dazu die `try-catch` - Blöcke).

Beispiel: Die Häufigkeit von Ziffern ermitteln (zahlen.java)

```
import java.io.*;

public class zahlen {

  public static void main (String args[]) {
   info();
   try {
     zaehlen();
   }
   catch (Exception e) {
    System.out.println("Datei nicht vorhanden");
   }
  }

  public static void info() {
    System.out.println("Die Häufigkeiten der Ziffern 0,1,2 ... 9 in
der derzeit größten bekannten Primzahl werden ermittelt:");
  }

  public static void zaehlen() throws Exception
  {
    File f = new File("prime5.txt");
    int ergebnis[] = new int[10];
    for (int i=0;i<ergebnis.length;i++)
```

Der SQL-Ausdruck der Abfrage `qryPartnerMann` liefert die Personendaten aus der Tabelle `tblPerson` und aus der Tabelle `tblPartner` dazu noch das Datum der Eheschließung.

```
SELECT tblPerson. ..., tblPartner.dtmVon
FROM tblPartner
INNER JOIN tblPerson
ON tblPartner.lMann = tblPerson.lPerson
WHERE
(((tblPartner.lFrau)=
[Forms]![frmHaupt]![lPerson]))
ORDER BY tblPerson.dtmGeburtstag;
```

In der Abfrage `qryPartnerFrau` werden nur die Felder `lMann` und `lFrau` der Partnerschaftstabelle ausgewechselt.

```
SELECT tblPerson. ..., tblPartner.dtmVon
FROM tblPartner
INNER JOIN tblPerson
ON tblPartner.lFrau = tblPerson.lPerson
WHERE
(((tblPartner.lMann)=
[Forms]![frmHaupt]![lPerson]))
ORDER BY tblPerson.dtmGeburtstag;
```

Zuordnung der Wohnorte

Die Zuordnung der Wohnorte ist prinzipiell gleich wie die Zuordnung von Partnern. Sie ist einfacher, weil lediglich die zentrale Person im Spiel ist und der Vorgang vom Geschlecht unabhängig ist.

Mit der Schaltfläche ***Wohnorte*** wird das Formular `frmWohnort` geöffnet. Beim Laden werden die Datensätze, die den Schlüssel der zentralen Person enthalten, ausgefiltert.

```
Private Sub Form_Load()
  DoCmd.ApplyFilter ,
        "lPerson = Forms!frmhaupt!lPerson"
```

Die Adressen, die zugeordnet werden sollen, müssen zuerst in der Tabelle `tblAdresse` gespeichert sein. Mit der Schalfläche ***Adressen*** öffnen Sie das Formular `frmAdresse` und dort können Sie sie eintragen. Die bereits gespeicherten Adressen können dann auf dem Formular `frmWohnort` mit der Kombobox `cboAdresse` der zentralen Person zugeordnet werden. Dabei wird der Adressenschlüssel `lAdresse` in die Tabelle `tblWohnort` gespeichert. In zwei Textfeldern unter der Kombobox können Sie die Gültigkeit des jeweiligen Wohnortes einschränken.

Anzeige der Wohnorte

Alle der zentralen Person zugeordneten Wohnorte werden auf dem Unterformular `fsubWohnort` in der Listbox `lstWohnort` aufgelistet. Die Listbox ist an die Abfrage `qryWohnort` gebunden.

Der SQL-Ausdruck der Abfrage `qryWohnort` liefert alle Datensätze aus der Tabelle `tblWohnort`, die den Schlüssel der zentralen Person enthalten und alle zugeordneten Adressangaben aus der Tabelle `tblAdresse`.

```
SELECT tblWohnort. ...,
tblAdresse. ...
FROM tblAdresse INNER JOIN tblWohnort
ON tblAdresse.lAdresse = tblWohnort.lAdresse
WHERE
(((tblWohnort.lPerson)=
[forms]![frmHaupt]![lPerson]))
ORDER BY tblWohnort.dtmVon DESC;
```

Ereignisse bearbeiten und anzeigen

Mit der Schaltfläche ***Ereignisse*** wird das Formular `frmEreignis` geöffnet. Beim Laden werden genauso wie bei Wohnorten die Datensätze, die den Schlüssel der zentralen Person enthalten, ausgefiltert.

Der SQL-Ausdruck für die Selektion der Ereignisse, welche in der Listbox `lstEreignis` auf dem Unterformular `fsubEreignis` angezeigt werden sollen, ist direkt in der Eigenschaft Datensatzherkunft eingetragen.

```
SELECT lPerson, sEreignis,dtmVon, dtmBis
FROM tblEreignis
WHERE lPerson=Forms!frmHaupt!lPerson
ORDER BY dtmVon
```

Schlusswort

Das Programm ist ein Beispiel dafür, dass eine gut durchdachte Datenmodellierung auch bei einer relativ komplexen Aufgabe zu einer einfachen und übersichtlichen Lösung führen kann.

alfred.nussbaumer@schule.at Alfred Nussbaumer

```
      ergebnis[i]=0;
    try {
      FileReader fr = new FileReader(f);
      BufferedReader eingabe = new BufferedReader(fr);
      String reihe = eingabe.readLine();
      long zaehler = 0;
      while (reihe != null) {
        for (int i=0; i<reihe.length();i++)
          for (int z=0; z<10; z++)
            if (((int) reihe.charAt(i) - 48) == z)
              ergebnis[z]++;
       reihe = eingabe.readLine();
       zaehler+=1;
       if ((zaehler % 100) == 0)
         System.out.print(".");
      }
    }
    catch (Exception e) {
     System.out.println("Datei nicht vorhanden");
    }

    for (int z=0; z<10;z++)
      System.out.print("\nHäufigkeit von " + z + ": " + ergebnis[z]);
  }
}
```

Beim Einlesen aus der festgelegten Datei liefert die Character-Stream-Klasse `FileReader` einen Datenstrom, der über die Character-Stream-Klasse `BufferedReader` zeilenweise gelesen wird. So lange die eingelesene Zeichenkette nicht „`null`“ ist, werden die einzelnen Bytes analysiert und die entsprechenden Zähler im Ergebnis-Array inkrementiert. Der Zeilenzähler `zaehler` dient lediglich dazu, jede 100. Eingabezeile einen Punkt auf der Konsole auszugeben.

Beispiel: Ziffernfolgen („Muster“) suchen (muster.java)

Wie im ersten Beispiel nachgeprüft werden kann, liefern große Primzahlen alle Ziffern etwa gleich oft. Eine reizvolle Aufgabe besteht nun darin, bestimmte Ziffernfolgen zu suchen. Deshalb verwenden wir in diesem Beispiel eine Datei, die die derzeit größte Primzahl als Folge von Bytes enthält (die Zeilenenden wurden entfernt). Die Byte-Stream-Klasse `RandomAccessFile` erlaubt den Zugriff auf jedes einzelne Byte.

```
import java.io.*;

public class muster {

  public static void main (String args []) throws IOException{
    char l=0;
    byte b;
    int anz=args[0].length();
    long pos=0;
    String vergl;
    RandomAccessFile datei = new RandomAccessFile("prime.txt","r");
    do {
      try {
        datei.seek(pos);
        vergl="";
        for (int j=0;j<anz;j++) {
          l=(char) datei.readByte();
          vergl=vergl+l;
        }
        if (vergl.equals(args[0]))
          System.out.println
            (args[0]+" an der "+pos+".ten Stelle gefunden!");
        pos++;
        if (pos % 100000 == 0)
          System.out.println("--->" + pos);
      }
      catch (EOFException e) {
        l=0;
      }
    } while (l != 0);
    datei.close();
  }
}
```

Die Methode `seek()` stellt den Dateizeiger zum Lesen (bzw. Schreiben) auf die in Bytes angegebene Stelle (`pos`). Mit der Methode `readByte()` wird das dort stehende Byte gelesen. Die gelesenen Bytes werden zu einem String verkettet, der schließlich mit dem beim Programmaufruf festgelegten Muster verglichen wird.

Die Ausgabe im letzten Beispiel lässt sich deutlich beschleunigen, wenn das Lesen und Vergleichen der Zeichenkette abgebrochen wird, sobald das erste Zeichen keine Übereinstimmung mehr ergibt:

```
import java.io.*;

public class muster_1 {

  public static void main (String args []) throws IOException{
    char l=0;
    byte b;
    int anz=args[0].length();;
    long pos=0;
    boolean ok;
    int i;
    String vergl;
    RandomAccessFile datei = new RandomAccessFile("prime.txt","r");
    do {
      try {
        datei.seek(pos);
        ok=true;
        i=0;
        do {
          l=(char) datei.readByte();
          if (l != args[0].charAt(i))
            ok = false;
          i++;
        } while (i<anz && ok);
        if (ok)
          System.out.println
            (args[0]+" an der "+pos+".ten Stelle gefunden!");
        pos++;
        if (pos % 100000 == 0)
          System.out.println("--->" + pos);
      }
      catch (EOFException e) {
        l=0;
      }
    } while (l != 0);
    datei.close();
  }
}
```

Anschließende Projekte für den Unterricht:

1. Das Bestimmen der Häufigkeiten von Zeichen hat eine interessante Bedeutung in der Kryptoanalyse: Mit einem ähnlichen Programm wie `zahlen.java` lässt sich die Häufigkeit von Zeichen in einem Geheimtext ermitteln. Dies lässt bei unsicheren Verfahren Rückschlüsse auf den verwendeten Verschlüsselungsalgorithmus zu (Beispiele zur Kryptographie werden in einem späteren Beitrag behandelt).

2. Für die zur Zeit längste Primzahl ergeben sich die folgenden Ziffernhäufigkeiten:

   Häufigkeit von 0: 405083
   Häufigkeit von 1: 405614
   Häufigkeit von 2: 405068
   Häufigkeit von 3: 405928
   Häufigkeit von 4: 405491
   Häufigkeit von 5: 404915
   Häufigkeit von 6: 405154
   Häufigkeit von 7: 405308
   Häufigkeit von 8: 406672
   Häufigkeit von 9: 404713

   Die Wahrscheinlichkeit für das Auftreten einer Ziffer ist demnach etwa 0,1. Das Programm `muster.java` soll so erweitert werden, dass die Häufigkeiten für die Ziffernfolgen 11, 111, 1111, etc. ermittelt werden. Stimmen die Ergebnisse mit den erwarteten Wahrscheinlichkeiten überein?

3. Für die obigen Programme sind GUI-Applikationen (unter der Verwendung von AWT- oder Swingkomponenten) zu erstellen. Die berechneten Häufigkeiten sind als Balkendiagramme darzustellen.