

Access-Tabellen vergleichen

Tabellen-Vergleichs-Tool COMPTAB

Karel Štípek

Es kommt ziemlich häufig vor, dass Sie den Inhalt von zwei Tabellen vergleichen brauchen.

- Daten aus verschiedenen Quellen werden zusammengefügt und dabei müssen doppelte Schlüsselwerte vermieden werden.
- Unterschiede zwischen Tabellen, die den Stand zu verschiedenen Zeitpunkten darstellen, werden ausgewertet.
- Bei der Entwicklung stellt der Vergleich der Daten vor und nach einer Programmänderung eine geeignete Testmethode dar. Sie können auch eine geänderte Abfrage in temporären Tabellen abspeichern und mehrere Versionen miteinander vergleichen.

Der Vorgang besteht grundsätzlich aus zwei Hauptaufgaben:

Hauptaufgaben eines Tabellenvergleichs

- Schlüssel-Felder werden verglichen. Dabei wird die Frage beantwortet, welche Datensätze in einer Tabelle vorhanden sind und in der anderen fehlen und umgekehrt.
- In den Datensätzen mit Schlüsselwerten, die in beiden Tabellen gleich sind, können auch die Nicht-Schlüssel-Felder miteinander verglichen werden.

Die oben genannten Aufgaben sind problemlos mit Abfragen zu lösen. Auch wenn die Arbeit mit der MS-Access-Oberfläche zum Abfra-

genentwurf durchaus einfach ist, stellt der auf diese Art und Weise durchgeführte Tabellenvergleich doch einen unnötigen Aufwand dar. Die zu erstellenden Abfragen sind meistens nur für einen einmaligen Gebrauch bestimmt und müssen daher nicht dauerhaft in der Applikation gespeichert werden.

Ein allgemein einsetzbares Tool, das die notwendigen Abfragen automatisch erstellt spart eine Menge Zeit.

Bedienung des Tools

Komfortable Benutzeroberfläche

Alle Bedienungselemente sind in einem einzigen Hauptformular untergebracht. Die linke und rechte Hälfte sind praktisch gleich und stellen die verglichenen Elemente der beiden Tabellen dar. Die Bedienung erfolgt logisch von der Datenbankauswahl oben bis zu der Ergebnisanzeige unten.

Die Schaltfläche mit dem Fragezeichen (links oben) öffnet ein Fenster mit einer Kurzanleitung.

Auswahl der Datenbanken

Die verglichenen Tabellen können in zwei unterschiedlichen oder in der gleichen Datenbank gespeichert werden. Die Schaltflächen Durchsuchen ermöglichen die Auswahl mit Hilfe des Standard-Öffnen-Dialogs. Der Pfad

und Name der ausgewählten Datenbank wird im darunter liegenden Textfeld angezeigt.

Die Schaltfläche DB1=DB2 überträgt die gleiche Datenbankangabe von der linken in die rechte Hälfte des Formulars ohne dass das Öffnen-Dialog angezeigt werden muss. Sie ist für den Fall bestimmt, dass beide Tabellen in der gleichen Datenbank liegen.

Verglichene Tabellen

Nachdem die Datenbank ausgewählt wird, wird automatisch die Combobox mit allen vorhandenen Tabellennamen befüllt und heruntergeklappt. Die verglichenen Tabellen lassen sich auswählen. Es muss sich dabei nicht um zwei unterschiedliche Tabellen handeln, auch ein Vergleich von verschiedenen Feldern der gleichen Tabelle ist problemlos möglich. Die Verknüpfung zwischen den Schlüsselfeldern muss dabei definiert werden auch wenn die Anzeige der fehlenden Datensätze in diesem Fall sinnlos ist.

Nach der Tabellenauswahl werden die darunter liegenden Listboxen mit allen vorhandenen Feldnamen befüllt.

Feldauswahl allgemein

Die Tabellenfelder können mit Hilfe von Schaltflächen mit Symbolen >, < und >> von den äußeren in die inneren Listbox übertragen

Abbildung -1: Hauptformular des Programms

werden. Dabei bleiben sie (im Gegensatz zum Standard-Verhalten in diversen Access-Assistenten) in den äußeren Listboxen erhalten. So ist es theoretisch möglich (auch wenn wahrscheinlich nicht oft gefragt), dass ein Feld sowohl als ein Schlüssel-, als auch als ein Nicht-Schlüssel-Feld eingesetzt wird, damit sein Wert mit einem anderen Feld verglichen werden kann.

Die Inhalte der inneren Listboxen werden beim Erstellen der Ergebnisabfragen berücksichtigt. Die Reihenfolge der Felder ist dabei wichtig. Es werden immer die gegenüberliegenden Felder miteinander verglichen. Dabei ist zu beachten, dass der Datentyp der zugehörigen Felder gleich sein muss.

Verknüpfte und verglichene Felder

Die Felder, die in die oberen beiden Listboxen *Verknüpfte Schlüsselfelder* übertragen werden, bilden die Verknüpfung (JOIN) zwischen den Tabellen.

Die unteren zwei Listboxen *Verglichene Felder* enthalten die Felder, die in den Datensätzen verglichen werden, bei denen die Inhalte der Schlüsselfelder gleich sind.

Anzeige der Ergebnisse

Die Schaltflächen im unteren Teil des Formulars zeigen alle möglichen Ergebnisabfragen an. Sie werden aktiviert oder deaktiviert je nach der Anzahl der ausgewählten Felder beider Tabellen. Die Strukturen der Abfragen werden aus der Auswahl nach folgendem Schema gebildet:

- **Tabelle 1**, bzw. **Tabelle 2** enthalten alle verknüpften und verglichenen Felder der ersten, bzw. der zweiten Tabelle
- **Fehlende** in der **Tabelle 1**, bzw. **Tabelle 2** enthalten alle verknüpften und verglichenen Felder der Datensätze, deren Schlüsselwerte in der anderen Tabelle nicht vorkommen.
- **Gleiche Werte** enthalten alle verknüpften und verglichenen Felder der Datensätze einer Tabelle, die alle gleich mit den Feldern der anderen Tabelle sind.
- **Unterschiedliche Werte** enthalten die verknüpften Felder einer Tabelle und die verglichenen Felder beider Tabellen. Es werden die Datensätze angezeigt, die mindestens in einem Feld unterschiedlich sind.

Makro Tastaturbelegung

Für die Erhöhung der Bedienungskomforts sind im Makro *Tastaturbelegung* drei Tastenkombination (Hot-Keys) definiert:



Tastenkombination	Aktion
	Fenster maximieren
	Fensterwiederherstellen

Tabelle -1: Implementierte Hot-Keys

Auswahl der Datenbanken und Tabellen

Programmstart

Zur Auswahl der Datenbanken wird im Programm das ActiveX-Steuerelement `CommonDialog` eingesetzt. Das Makro `AutoExec` startet die Funktion `ProgrammStart()`, die die Referenz zu diesem Objekt überprüft.

```
...
On Error GoTo Err_ProgrammStart
Set ref = Application.References("MSComDlg")
...
Err_ProgrammStart:
...
```

```
g.OcXused = False
```

Sollte das ActiveX-Steuerelement nicht richtig registriert sein, wird eine Meldung ausgegeben und die globale Variable `OcXused` auf `False` gesetzt.

Hauptformular öffnen

Beim Laden des Hauptformulars wird die Prozedur `FormInit` für beide Formularhälften aufgerufen. Sollte das ActiveX-Steuerelement nicht verfügbar sein, werden die Schaltflächen Durchsuchen deaktiviert. In diesem Fall müssen für die Auswahl der Datenbanken ihre Pfade und Namen in den Textfeldern manuell eingegeben werden.

```
Sub Form_Load()
FormInit "1"
FormInit "2"
Me.cmdMDB1.Enabled = g.OcXused
Me.cmdMDB2.Enabled = g.OcXused
```

Prozedur FormInit

Die Prozedur wird mit einem Parameter aufgerufen, damit sie auch nur für eine Formularhälfte eingesetzt werden kann. Die Verknüpfung zu der verglichenen Tabelle wird gelöscht und die Steuerelemente für die Tabellen- und Feldauswahl initialisiert. Für das Löschen der Inhalte der inneren Listboxen werden die Ereignis-Prozeduren der Schaltflächen "<<" bzw. ">>" aufgerufen.

```
Sub FormInit(t$)
TableDelete "T" + t
Me.Controls("cboTab" + t) = Null
Me.Controls("1stFeld" + t).Requery
If t = "1" Then
cmdJoin1Init_Click
cmdComp1Init_Click
Else
cmdJoin2Init_Click
...
```

Prozedur TableDelete

`TableDelete` ist eine fehlertolerante Prozedur, die eine Tabelle oder eine Tabellenverknüpfung in der aktuellen Datenbank löscht.

```
Sub TableDelete(Name$)
Dim tdf As TableDef
For Each tdf In CurrentDb.TableDefs
If tdf.Name = Name Then
CurrentDb.TableDefs.Delete Name
Exit Sub
End If
Next tdf
End Sub
```

Datenbank öffnen

Die meisten Selektionsvorgänge sind für beide Tabellen (Formularhälften) identisch und werden aus diesem Grund nur für die erste Tabelle (linke Hälfte) erklärt.

Die Schaltfläche *Durchsuchen* ruft die Funktion `GetDatabaseName()` auf, die das Dialog öffnet und den ausgewählten Pfad und Datenbanknamen in das Textfeld `txtMDB1` schreibt. Durch den Aufruf der Ereignisprozedur `AfterUpdate` des Textfeldes wird der weitere Programmablauf gleich wie nach der manuellen Eingabe in das Textfeld.

```
Sub cmdMDB1_Click()
txtMDB1 = GetDatabaseName()
txtMDB1_AfterUpdate
End Sub
```

Funktion GetDatabaseName

Funktion `GetDatabaseName()` öffnet das ActiveX-Steuerelement `CommonDialog`, das im Steuerelement `ocxOpen` untergebracht ist und liefert den ausgewählten Pfad und Datenbanknamen.

```
Function GetDatabaseName()
With Me.ocxOpen
```

```
.DialogTitle = "Wählen Sie die Datenbank aus"
.Filter = "Access Datenbanken (*.mdb)|*.mdb|MDE Dateien (*.mde)|*.mde|"
.FileName = ""
.ShowOpen
GetDatabaseName = .FileName
...
```

Auswahl von Tabellen

Nachdem eine Datenbank im Öffnen-Dialog oder durch die manuelle Eingabe im Textfeld ausgewählt wurde, wird die Combobox `cboTab1` für die Auswahl von Tabellen befüllt und heruntergeklappt. Sein Inhalt wird der Systemtabelle `MSysObjects` entnommen. Die Einträge mit `Type=1`, die nicht mit "MSys" beginnen, stellen die normalen Tabellen dar.

```
Sub txtMDB1_AfterUpdate()
...
cboTab1.RowSource = "SELECT name FROM [" & txtMDB1 & "].msysobjects WHERE (type=1) and not name like 'MSys*' ORDER BY name;"
Me.cboTab1.Enabled = True
Me.cboTab1.SetFocus
Me.cboTab1.DropDown
...
```

Tabellen verknüpfen – Prozedur TabConnect

Nach der Auswahl der Tabelle in der Combobox `cboTab1` wird die Prozedur `TabConnect` aufgerufen, die die Verknüpfung zu der gewünschten Tabelle erstellt. Die Verknüpfungen bekommen die Namen "T1", bzw. "T2". Mit diesen Namen werden die Tabellen in den generierten Abfragen referenziert. Die Prozedur wird mit dem gleichen Parameter wie `FormInit` aufgerufen, der die erste oder zweite Tabelle bezeichnet. Am Ende der Prozedur wird die Listbox, die die verfügbaren Tabellenfelder anzeigt, initialisiert. Dieses Steuerelement muss nicht näher erklärt werden, die notwendige Funktionalität wird nur durch das Einstellen der Eigenschaften `Herkunftstyp` auf Feldliste und `Datenherkunft` auf die Tabelle T1, bzw. T2 erreicht.

```
Sub TabConnect(t$)
...
TableDelete "T" + t
Set tdf = db.CreateTableDef("T" + t)
tdf.Connect = ";Database=" + Me.Controls("txtMDB" + t)
tdf.SourceTableName = Me.Controls("cboTab" + t)
db.TableDefs.Append tdf
Me.Controls("1stFeld" + t).Requery
...
```

Auswahl der verknüpften und verglichenen Felder

Innere Listboxen

Auf dem Formular befinden sich vier innere Listboxen. In diesen werden die verknüpften und verglichenen Felder beider Tabellen angezeigt. Die ausgewählten Feldnamen werden in Datenfeldern (Arrays) gespeichert. Pro jede Listbox wird ein Array definiert.

Für das Erstellen der Ergebnisabfragen sind eigentlich die Inhalte der Arrays relevant. Die Listboxen werden dabei nicht abgefragt und stellen nur die Präsentation der Auswahl dar.

Namen-Übersicht

Die vier Listboxen heißen `1stJoin1` und `1stJoin2` für die verknüpften und `1stComp1` und `1stComp2` für die verglichenen Felder. Die gleiche Funktionalität ist viermal vorhanden, aus diesem Grund werden die Programmteile nur einmal für die Listbox `1stJoin1` beschrieben. Zuerst die

Übersicht der Namen der zusammenhängenden Objekte:

Objekt	Bedeutung
1stJoin1	innere Listbox
arrJoin1	zugehöriges Array
cmdJoin1on	Feld(er)einfügen
cmdJoin1off	Feld(er)entfernen
cmdJoin1init	alle Feld(er)entfernen

Tabelle -2: Objekte für die Auswahl der verknüpften Felder der ersten Tabelle

Funktion FillLstFromArray()

Die Ereignis-Prozeduren der drei o.g. Schaltflächen pro Listbox müssen nur die Arrays richtig befüllen. Die Anzeige der Inhalte der Arrays in den Listboxen erfolgt automatisch durch die Funktion FillLstFromArray(). Ihr Name wird gleich für alle vier Listboxen in der Eigenschaft Herkunftstyp eingetragen. Die Funktion wird an dieser Stelle nicht ausführlich beschrieben, weil sie grundsätzlich der Standardlösung entspricht, die in der Hilfe unter "RowSourceType-Eigenschaft" zu finden ist.

Felder einfügen - Funktion FillSelArr()

Die Funktion FillSelArr() übernimmt die markierten Einträge (Feldnamen) der äußeren Listbox in das entsprechende Array der inneren Listbox. Sie wird beim Aktivieren zum Beispiel der Schaltfläche cmdJoin1on mit zwei Parametern aufgerufen:

- 1st die äußere Listbox
- arr Array der inneren Listboxen

Bei jedem Aufruf wird in allen Elementen der äußeren Listbox die Eigenschaft Selected überprüft und die markierten Feldnamen an Ende des Arrays angehängt.

```
Function FillSelArr(1st As Control,
    arr As Variant)
...
For i = 0 To 1st.ListCount - 1
    If 1st.Selected(i) Then
        If Len(arr(0)) > 0 Then
            ReDim Preserve arr(UBound(arr) + 1)
        End If
        arr(UBound(arr)) = 1st.Column(0, i)
    End If
Next i
...
```

Felder entfernen - Funktion ArrElemDelete()

Die Funktion ArrElemDelete() wird aufgerufen, wenn die Schaltfläche zum Entfernen von ausgewählten Feldern aktiviert wird. Sie hat zwei Parameter, die ähnliche Bedeutung haben, wie bei FillSelArr(). Diesmal wird aber nur die innere Listbox behandelt.

- 1st die innere Listbox
- arr Array der inneren Listboxen

Die Elemente der inneren Listboxen werden vom Ende auf die Eigenschaft Selected überprüft und für die markierten wird die Funktion zum Entfernen eines Array-Elements OneElemDelete() aufgerufen.

```
Sub ArrElemDelete(1st As Control,
    arr As Variant)
For i = 1st.ListCount - 1 To 0 Step -1
    If 1st.Selected(i) Then
        OneElemDelete arr, i
    End If
Next i
```

Funktion OneElemDelete()

Die Funktion ArrElemDelete() hat zwei Parameter:

- arr Array der inneren Listboxe

● delind Index des gelöschten Elements

Zuerst werden alle Elemente ab dem gelöschten um eine Position zum Anfang verschoben und dann die Dimension des Arrays um 1 verringert.

Dabei gibt es ein kleines Problem – wenn alle ausgewählten Felder entfernt werden, muss im Array doch ein Element mit dem Index 0 erhalten bleiben. In dem Fall wird das Array nicht redimensioniert, sondern das erste Element geleert. Die Länge dieses ersten Elements wird beim Erstellen der Ergebnisabfragen überprüft, um festzustellen, ob mindestens ein Feld für den jeweiligen Teil des SQL-Ausdrucks ausgewählt wurde.

```
Sub OneElemDelete(arr As Variant, delind%)
...
For i = delind To UBound(arr) - 1
    arr(i) = arr(i + 1)
Next i
If UBound(arr) > 0 Then
    ReDim Preserve arr(UBound(arr) - 1)
Else
    arr(0) = ""
End If
...
```

Alle Felder entfernen

Wenn alle ausgewählten Felder aus der inneren Listbox entfernt werden sollen, reicht es einfach, nur das zugehörige Array ohne Preserve zu redimensionieren. Damit wird es vollständig initialisiert und die innere Listbox geleert.

```
Sub cmdJoin1Init_Click()
    ReDim arrJoin1(0)
    ...
```

Nach jeder Änderung der Feldauswahl

In der Ereignisprozedur jeder Schaltfläche, die die Feldauswahl beeinflusst (insgesamt 12 Stück), müssen noch zwei Aktionen durchgeführt werden.

- die entsprechende innere Listbox wird neu angezeigt, damit die Veränderung sichtbar ist
- die Funktion ActionEnable() wird aufgerufen, damit die Schaltflächen zur Anzeige der Vergleichsergebnisse entsprechend der Auswahl aktiviert oder deaktiviert werden

So schaut also beispielweise die komplette Ereignisprozedur der Schaltfläche zur Auswahl eines verknüpften Schlüsselfeldes der ersten Tabelle aus:

```
Sub cmdJoin1on_Click()
    FillSelArr 1stFeld1, arrJoin1
    1stJoin1.Requery
    ActionEnable
End Sub
```

Erstellen und Anzeigen von Ergebnisabfragen

Übersicht der Schaltflächen und Abfragen

Wenn die Arrays mit ausgewählten Feldern richtig befüllt sind, können die Ergebnisse des Tabellenvergleichs ausgewertet werden.

Die dynamisch erstellten Ereignisabfragen haben die gleichen Namen wie die Aufschriften auf den im unteren Teil des Formulars angebrachten Schaltflächen:

Prozedur ActionEnable()

Das Erstellen von Abfragen ist nur dann sinnvoll, wenn auch die notwendigen Felder ausgewählt sind und die Anzahl der Felder in beiden verglichenen Tabellen gleich sind. Durch das Aktivieren und Deaktivieren von Ergebnis-Schaltflächen wird der Anwender auf die

Schaltfläche	Abfrage
cmdTab1	Tabelle 1
cmdTab2	Tabelle 2
cmdTab1more	Fehlende in der Tabelle 2
cmdTab2more	Fehlende in der Tabelle 1
cmdFeldEq	Gleiche Werte
cmdFeldDiff	Unterschiedliche Werte

Tabelle -3: Schaltflächen und Abfragen für die Anzeige der Ergebnisse

unvollständige Auswahl sofort aufmerksam gemacht.

Die Aufgabe wird mit der Prozedur ActionEnable() gelöst, die in der Ereignisprozedur jeder Schaltfläche, die die Feldauswahl beeinflusst, aufgerufen wird. In der Funktion muss sowohl die Anzahl der Array-Elemente als auch die Länge des ersten Elements überprüft werden (siehe oben).

```
j1 = UBound(arrJoin1)
... j2, c1, c2 dtto
ej1 = (Len(arrJoin1(0)) = 0)
... ej2, ec1, ec2 dtto
Me.cmdTab1.Enabled = Not ej1 Or Not ec1
Me.cmdTab1more.Enabled =
    Not ej1 And Not ej2 And (j1 = j2)
Me.cmdFeldDiff.Enabled =
    Me.cmdTab1more.Enabled
    And Not ec1 And Not ec2 And (c1 = c2)
```

Ergebnisabfragen allgemein

Für die Anzeige jedes Ergebnisses des Tabellenvergleichs wird der gleiche Vorgang angewendet:

- aus den Feldnamen, die in den Selektion-Arrays gespeichert sind, wird der notwendige SQL-Ausdruck erstellt
- dieser SQL-Ausdruck wird der gewünschten Abfrage zugewiesen
- die Abfrage wird sofort geöffnet

Prozedur QueryDefOpen

Die Prozedur QueryDefOpen erledigt die letzten zwei Aufgaben des oben angeführten allgemeinen Vorgangs. Sie wird mit zwei Parametern aufgerufen:

- qryname Name der Abfrage
- sqlstr der SQL-Ausdruck

In dieser Funktion ist auch eine Fehlerbehandlung implementiert. Nachdem die Abfrage nicht ausgeführt werden kann (z.B. bei inkompatiblen Datentypen der verglichenen Felder), wird die Access-interne Fehlermeldung angezeigt.

```
Sub QueryDefOpen(qryname$, sqlstr$)
On Error GoTo Err_QueryDefOpen
Dim qdf As QueryDef
Set qdf = CurrentDb.QueryDefs(qryname)
qdf.SQL = sqlstr
qdf.Close
DoCmd.OpenQuery (qryname)
Exit_QueryDefOpen:
Exit Sub
Err_QueryDefOpen:
MsgBox Error
Resume Exit_QueryDefOpen
End Sub
```

Teilfunktionen für das Erstellen von SQL-Ausdrücken

Damit das Erstellen vom SQL-Ausdruck für einzelne Aufgaben des Vergleichs weitgehend modularisiert werden kann, werden SQL-Teilfunktionen eingesetzt, die die Feldausdrücke aufgrund der Inhalte der Selektions-Arrays zusammen bilden. Zur Erinnerung: Die Abkürzung Join betrifft die verknüpften, Comp die verglichenen Felder.

Arrays bearbeiten

Die aus den Arrays übernommen Feldnamen werden in eckige Klammern eingeschlossen. Es ist für den Fall notwendig, wenn sie z.B. Leerzeichen enthalten, damit es nicht zu einem Syntaxfehler in SQL kommt.

Die Array-Elemente werden in einer Schleife abgearbeitet und mit Komma getrennt, bzw. die einzelnen Vergleichsoperationen mit dem logischen Operator **AND** oder **OR** verbunden. Für die richtige Syntax des gesamten SQL-Ausdruck wird vor dem Verlassen der Funktion aus dem gebildeten Ausdruck das letzte Zeichen oder Wort weggeschnitten.

Die Rückgabewerte dieser Teilfunktionen werden dann in der Ereignis-Prozedur `onClick` der jeweiligen Schaltfläche mit Hilfe der zusätzlichen SQL-Anweisungen zum fertigen SQL-Ausdruck der Ergebnisabfrage zusammengefasst und an die Funktion `QueryDefOpen()` als Parameter übergeben.

Funktion SQL_JoinComp()

Die Funktion `SQL_JoinComp()` liefert eine Auflistung der Feldnamen der verknüpften und verglichenen Felder einer Tabelle. Diese Auflistung wird als Teil der `SELECT`-Anweisung verwendet. Die Nummer der Tabelle (1 oder 2) wird als Parameter `tabnum` übergeben.

```
Function SQL_JoinComp(tabnum%) As String
```

```
...
If tabnum = 1 Then
    arrj = arrJoin1
    arrc = arrComp1
Else
    ...
If Len(arrj(0)) > 0 Then
    For i = 0 To UBound(arrj)
        s = s & "T" & CStr(tabnum) &
            "." & Trim$(arrj(i)) & ","
    Next i
End If
If Len(arrc(0)) > 0 Then
    ...
SQL_JoinComp = Left$(s, Len(s) - 1)
```

Funktion SQL_Joins()

Die Funktion `SQL_Joins()` liefert eine Auflistung der verknüpften Felder beider Tabellen. Sie wird im `ON`-Ausdruck der `JOIN`-Anweisung eingesetzt.

```
Function SQL_Joins() As String
```

```
For i = 0 To UBound(arrJoin1)
    s = s & "(T1.[ " & Trim$(arrJoin1(i)) & "]="
        & "T2.[ " & Trim$(arrJoin2(i)) & "] ) AND "
Next i
SQL_Joins = Left$(s, Len(s) - 5)
```

Funktion SQL_Comp12()

Die Funktion `SQL_Comp12()` liefert eine Auflistung der verknüpften Felder der ersten Tabelle und der verglichenen Felder beider Tabellen. Sie wird in der `SELECT`-Anweisung der Abfrage Unterschiedliche Werte eingesetzt.

```
Function SQL_Comp12()
```

```
...
```

```
For i = 0 To UBound(arrJoin1)
    s = s & "T1.[ " & Trim$(arrJoin1(i)) & "], "
Next i
For i = 0 To UBound(arrComp1)
    s = s & "T1.[ " & Trim$(arrComp1(i)) & "], "
        & "T2.[ " & Trim$(arrComp2(i)) & "], "
Next i
SQL_Comp12 = Left$(s, Len(s) - 1)
```

Funktion SQL_FieldEq()

Die Funktion `SQL_FieldEq()` liefert den Ausdruck für das Finden der gleichen Inhalte der verglichenen Felder beider Tabellen. Die Vergleichsoperationen werden in diesem Fall mit dem logischen Operator `AND` verbunden. Die Funktion wird im `WHERE`-Ausdruck der Abfrage `Gleiche Werte` eingesetzt.

Wenn beide verglichenen Felder `NULL` enthalten, werden sie auch als gleiche Werte behandelt.

```
Public Function SQL_FieldEq() As String
```

```
...
For i = 0 To UBound(arrComp1)
    s = s & "(T1.[ " & Trim$(arrComp1(i))
        & "]=" & "T2.[ " & Trim$(arrComp2(i))
        & "] ) OR "
    s = s & "(IsNull(T1.[ "
        & Trim$(arrComp1(i)) & "] ) AND "
    s = s & "IsNull(T2.[ "
        & Trim$(arrComp2(i)) & "] ) ) AND "
Next i
SQL_FieldEq = Left$(s, Len(s) - 5)
```

Funktion SQL_FieldDiff()

Die Funktion `SQL_FieldDiff()` liefert den Ausdruck für das Finden der Unterschiede der verglichenen Felder beider Tabellen. Die Vergleichsoperationen werden in diesem Fall mit dem logischen Operator `OR` verbunden. Die Funktion wird im `WHERE`-Ausdruck der Abfrage `Unterschiedliche Werte` eingesetzt.

Wenn nur eines der Felder `NULL` enthält, werden sie auch als nicht gleiche Werte behandelt.

```
Function SQL_FieldDiff() As String
```

```
...
For i = 0 To UBound(arrComp1)
    s = s & "(T1.[ " & Trim$(arrComp1(i))
        & "]<>" & "T2.[ " & Trim$(arrComp2(i))
        & "] ) OR "
    s = s & "(IsNull(T1.[ " & Trim$(arrComp1(i))
        & "] )<>"
        & "IsNull(T2.[ " & Trim$(arrComp2(i))
        & "] ) ) OR "
Next i
SQL_FieldDiff = Left$(s, Len(s) - 4)
```

Komplette SQL-Ausdrücke in den Ergebnisabfragen

Tabelle...

Die Abfragen `Tabelle1` und `Tabelle 2` zeigen alle ausgewählten Felder (zuerst die Schlüsselfelder) einer Tabelle an. Die Ereignis-Prozedur der Schaltfläche `cmdTab1` hat die einfachste mögliche Form.

```
Sub cmdTab1_Click()
```

```
    QueryDefOpen "Tabelle 1",
        "SELECT " & SQL_JoinComp(1) & " FROM T1"
```

Fehlende in der Tabelle...

Der SQL-Ausdruck für diese und weitere Abfragen besteht aus den Rückgabewerten mehrerer SQL-Teilfunktionen und wird zuerst in der Variablen `s` zusammengesetzt, bevor er an die Funktion `QueryDefOpen()` übergeben wird.

Um die fehlenden Datensätze (Schlüsselwerte) zu finden, wird in der `WHERE`-Anweisung der Wert des ersten verknüpften Feldes der anderen Tabelle auf den Wert `NULL` überprüft.

```
Sub cmdTab1more_Click()
```

```
...
s = "SELECT DISTINCT " & SQL_JoinComp(1)
    & " FROM T1 LEFT JOIN T2"
    & " ON " & SQL_Joins()
    & " WHERE T2.[ " & arrJoin2(0) & " ] IS NULL"
    QueryDefOpen "Fehlende in der Tabelle 2", s
```

Gleiche Werte

Bei einem positiven Ergebnis dieses Vergleichs sind sowohl alle verknüpften als auch alle verglichenen Felder beider Tabellen gleich. Aus diesem Grund werden nur die Felder der ersten Tabelle mit der Teilfunktion `SQL_JoinComp(1)` angezeigt.

Den Verknüpfungsausdruck wird von der Funktion `SQL_Joins()` und der Filterausdruck von der Funktion `SQL_FieldEq()` geliefert.

```
Private Sub cmdFieldEq_Click()
```

```
...
s = "SELECT DISTINCT "
    & SQL_JoinComp(1)
    & " FROM T1 INNER JOIN T2"
    & " ON " & SQL_Joins()
    & " WHERE " & SQL_FieldEq()
    QueryDefOpen "Gleiche Werte", s
```

Unterschiedliche Werte

In dieser Abfrage werden die verknüpften Felder der ersten Tabelle und die verglichenen Felder beider Tabellen mit der Funktion `SQL_Comp12()` angezeigt.

Den Verknüpfungsausdruck bleibt gleich wie in der Abfrage `Gleiche Werte` und der Filterausdruck wird von der Funktion `SQL_FieldDiff()` geliefert.

```
Private Sub cmdFieldDiff_Click()
```

```
...
s = "SELECT DISTINCT "
    & SQL_Comp12()
    & " FROM T1 INNER JOIN T2"
    & " ON " & SQL_Joins()
    & " WHERE " & SQL_FieldDiff()
    QueryDefOpen "Unterschiedliche Werte", s
```

Schlusswort

Die mit dem vorgestellten Tool dynamisch erstellten Abfragen bleiben in der Datenbank `CompTab.mdb` solange gespeichert, bis ein neuer Vergleich durchgeführt wird. Sie stehen wie alle anderen Abfragen im Datenbankfenster zur Verfügung. Man kann sie unter einem anderen Namen speichern, in andere Datenbanken exportieren, verändern und beliebig weiter verwenden.

Dateien zum Artikel





Die zugehörigen Dateien finden Sie bei der Online-Version zu diesem Artikel: <http://pcnews.at/?id=pcn97002200>. Die Online-Version kann ganz allgemein systematisch oder direkt über die Seitennummer gefunden werden:

Systematisch

<http://pcnews.at/> -> *Inside* -> *Ausgaben* -> *Nummer der Ausgabe* -> *Artikel*

Direkt

<http://pcnews.at?id=pcn<n><s><i>>, **n** Heftnummer ohne führende Nullen; **s** Seite mit führenden Nullen, 4-stellig; **i** fortlaufende Nummer für mehrere Artikel auf einer Seite, 2-stellig mit führenden Nullen, beginnend bei **00**.

Grundsätzlich wird die *Pdf-Version*  angezeigt, und auch - falls verfügbar - die *Html-Version* , *ergänzende Dateien*  und *Links* .