

Berichte zur Laufzeit filtern

Die Datenquelle eines Berichts verhält sich gemeinhin statisch. Die Auswahl der Daten geschieht entweder im Vorfeld über die Filterung einer Abfrage oder über Gruppierungsebenen des Berichts. Mit der in Access 2007 eingeführten Berichtsansicht jedoch kommt mehr Leben ins Spiel. Diese erlaubt zur Laufzeit eine Modifikation der zugrundeliegenden Daten. Der folgende Beitrag zeigt zudem, wie Sie das Layout über Bedingte Formatierung von Textfeldern aufpeppen können.

Beispieldatenbank

Die Beispiele dieses Artikels finden Sie in der Datenbank **1701_BerichtsFilter.accdb**.

Die Berichtsansicht

Ein Bericht kommt üblicherweise für den gestalteten Ausdruck von Daten zur Anwendung. Wollten Sie dabei mehrere Auszüge der gleichen Daten mit unterschiedlichen Bedingungen ausdrucken, so war dies einst ein umständlicheres Unterfangen. Die Auswahl der Daten hatte etwa über eine Abfrage stattzufinden, in der Sie die Bedingungen über Parameter von außen steuerten. Hier könnten Sie VBA-Funktionen verwenden, die statt fest eingebauter Parameter variable Vergleichswerte liefern.

Bei der **Berichtsansicht** benötigen Sie derlei Kniffe nicht. Sie ermöglicht die dynamische Filterung der Datenquelle zur Laufzeit, ähnlich, wie bei einem Formular. Sie schalten die Berichtsansicht entweder über die entsprechende Ribbon-Schaltfläche ein, oder Sie setzen sie gleich im Entwurf des Berichts, indem Sie die Eigenschaft **Standardansicht** auf **Berichtsansicht** einstellen. Diese Ansicht ist, im Gegensatz zur **Layoutansicht**, immer verfügbar und muss in den Optionen der Datenbank nicht ausdrücklich erlaubt werden. Wohl aber existiert eine Eigenschaft **Berichtsansicht zulassen** unter dem **Format-Reiter** des Eigenschaftensblatts, die auf **Ja** stehen muss, damit der Bericht zur Laufzeit in diese umgeschaltet werden kann. Sonst stehen nur die **Seitenansicht** und gegebenenfalls die **Layoutansicht** zur Verfügung.

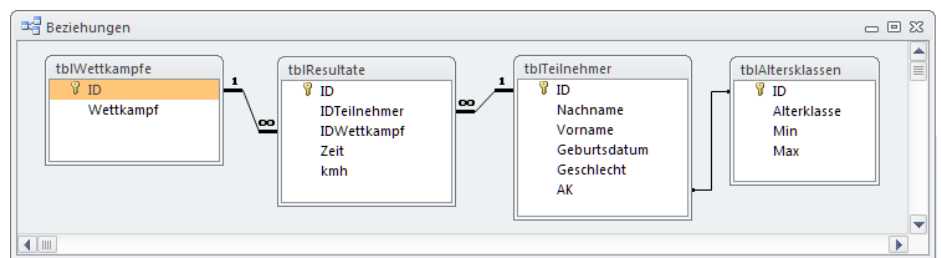


Bild 1: Das Datenmodell der Beispieldatenbank zu fingierten Wettkampfdaten

Tatsächlich verändert der Bericht in dieser Ansicht komplett sein Wesen und wird im Grunde zu einem Formular. Ein Fenster-Tool etwa zeigt, dass sich die verwendete Fensterklasse nun nicht mehr **OReport** nennt, sondern **OForm**, also identisch zur für Formularfenster verwendeten.

Für das Beispiel zu diesem Beitrag kommt eine Datenbank zum Einsatz, die bereits in der Ausgabe **5/2015, Daten ins Web exportieren**, zur Demonstration von Web-Exporten diente. Das Datenmodell für jene Wettkampfdatenbank wird hier nicht nochmals erläutert. Das Beziehungsmodell mit seinen vier Tabellen ist in Bild 1 gezeigt. Es gibt Radwettkämpfe, Teilnehmer in verschiedenen Altersklassen und deren Wettkampfergebnisse.

Ziel der aktuellen Beispieldatenbank ist nun nicht ein Web-Export, sondern der Ausdruck der Resultate über einen Bericht. Diese sollen wahlweise nach Wettkampfkategorien und Altersklassen filterbar sein. Das damals dazu verwendete Formular ist in der Datenbank ebenfalls enthalten (**frmErgebnisse**), aber um eine Spalte erweitert, die die Durchschnittsgeschwindigkeit visuell darstellt (siehe Bild 2). Dabei kommt für dieses Textfeld kein Grafiksteuerelement zum

Einsatz, sondern ein Textfeld mit **Bedingter Formatierung**, dessen Einstellungen wir gleich noch eingehender beleuchten.

Die beiden Kombinationsfelder oben dienen der Filterung der Resultate, wobei wir hierfür kleine VBA-Ereignisprozeduren benutzen. Das Formular baut direkt auf der Abfrage `qry_Resultate` auf und kommt im Endlosmodus daher.

Mit Klick auf die Schaltfläche **Berichtsversion...** gelangen Sie zu einem anderen Formular, das jedoch einen Unterbericht einbaut hat, der die Ergebnisse im Prinzip analog darstellt, auch wenn sich die Farbgebung hier etwas unterscheidet. Dieses Formular (siehe Bild 3) und seinen Unterbericht sehen wir uns genauer an.

Das umrahmende Formular ist selbst an keine Datenquelle gebunden und steuert lediglich den Unterbericht. Eben dies ist eine Neuerung seit **Access 2007**. Früher konnte man einen Unterbericht nicht als Steuerelement in ein Formular integrieren, sondern nur wiederum in Berichten. Diese Möglichkeit ist allerdings nur auf Unterberichte beschränkt, die die **Berichtsansicht** als Standard eingestellt haben. Die **Seitenansicht** kann nicht in ein Formular eingebettet werden.

AK	Geschlecht	Name	Geburtsdatum	Zeit	kmh	Geschwindigkeit visuell
Sen III	m	Abendroth, Margret	28.10.1956	01:49:47	30,058	[Bar]
Standard	m	Ahlborn, Jacoba	28.07.1988	01:23:58	39,300	[Bar]
Sen I	m	Albrecht, Gudrun	03.05.1983	01:49:53	30,032	[Bar]
Sen II	m	Aldenhoff, Susanne	23.12.1970	01:28:01	37,495	[Bar]
Sen I	m	Allen, Regina	19.08.1981	01:32:18	35,752	[Bar]
Sen IV	m	Allgeier, Iris	14.10.1950	01:48:06	30,525	[Bar]
Sen IV	m	Allmendinger, Elke	18.07.1952	01:34:48	34,811	[Bar]
Sen I	m	Alte, Ingrid	29.12.1981	01:43:12	31,974	[Bar]
Sen IV	m	Amann, Christa	09.11.1952	01:40:02	32,988	[Bar]
U21	m	Amershuber, Christine	21.12.1995	01:39:13	33,259	[Bar]
Sen II	m	Ammer, Heidemarie	22.01.1966	01:40:43	32,768	[Bar]
Standard	m	Andres, Ines	21.05.1994	01:28:18	37,375	[Bar]
Sen IV	m	Arnold, Grit	30.04.1954	01:32:44	35,584	[Bar]
Standard	m	Auffenberg, Gisela	03.03.1994	01:31:07	36,219	[Bar]
Standard	m	Augustin, Simone	20.07.1991	01:49:53	30,032	[Bar]
Sen II	m	Auserkorte, Ruth	29.04.1974	01:36:08	34,329	[Bar]
Sen I	m	Bech, Emmy	04.04.1985	01:40:07	32,962	[Bar]

Bild 2: Das Formular `frmErgebnisse` zur Auswertung der Wettkampfergebnisse

AK	Geschlecht	Name	Geburtsdatum	Zeit	kmh	Geschwindigkeit visuell
Sen III	w	Sechelmann, Werner	07.04.1960	03:24:48	43,944	[Bar]
Sen III	w	Terrahe, Michael	20.06.1962	03:27:02	43,471	[Bar]
Sen III	w	Oldenburg, Andreas	24.06.1959	03:27:59	43,273	[Bar]
Sen III	w	Henke, Heinrich	29.11.1959	03:28:53	43,086	[Bar]
Sen III	w	Schüppel, Ekkehard	30.11.1961	03:32:27	42,364	[Bar]
Sen III	w	Plöhn, Hanns-Joachim	12.12.1961	03:33:15	42,204	[Bar]
Sen III	w	Keppeler, Dominik	08.03.1965	03:37:58	41,291	[Bar]
Sen III	w	Wolf, Michael	14.07.1956	03:43:38	40,245	[Bar]
Sen III	w	Markowsky, Bernd	30.04.1965	03:46:10	39,793	[Bar]
Sen III	w	Thomas, Walter	04.12.1957	03:56:04	38,126	[Bar]
Sen III	w	Dechert, Manfred	20.08.1963	03:57:11	37,946	[Bar]
Sen III	w	Bünger, Michael	29.08.1962	03:59:44	37,541	[Bar]
Sen III	w	Will, Klaus-Peter	14.12.1964	04:02:51	37,060	[Bar]
Sen III	w	Hoffmann, Georg	14.05.1957	04:04:33	36,801	[Bar]
Sen III	w	Willems, Willi	14.05.1965	04:07:15	36,402	[Bar]
Sen III	w	Kratsch, Rudolf	20.07.1962	04:09:25	36,085	[Bar]
Sen III	w	Walsemann, Michael	05.01.1956	04:10:03	35,992	[Bar]
Sen III	w	Fahlbusch, Rainer	30.11.1957	04:20:30	34,549	[Bar]

Bild 3: Hier haben wir es nicht mehr mit einem Formular in Endlosansicht zu tun, sondern mit einem Unterbericht im Modus **Berichtsansicht** (`frmErgebnisseReport`)

AK	Gesch	Name	Geburtsdatum	Zeit	kmh	kmh
1						

Bild 4: Der Entwurf des Unterberichts für das Formular `frmErgebnisseReport`

Den recht einfachen Entwurf des Berichts `rptErgebnisse`, der nur einen Detailbereich enthält, finden Sie in Bild 4 dargestellt.

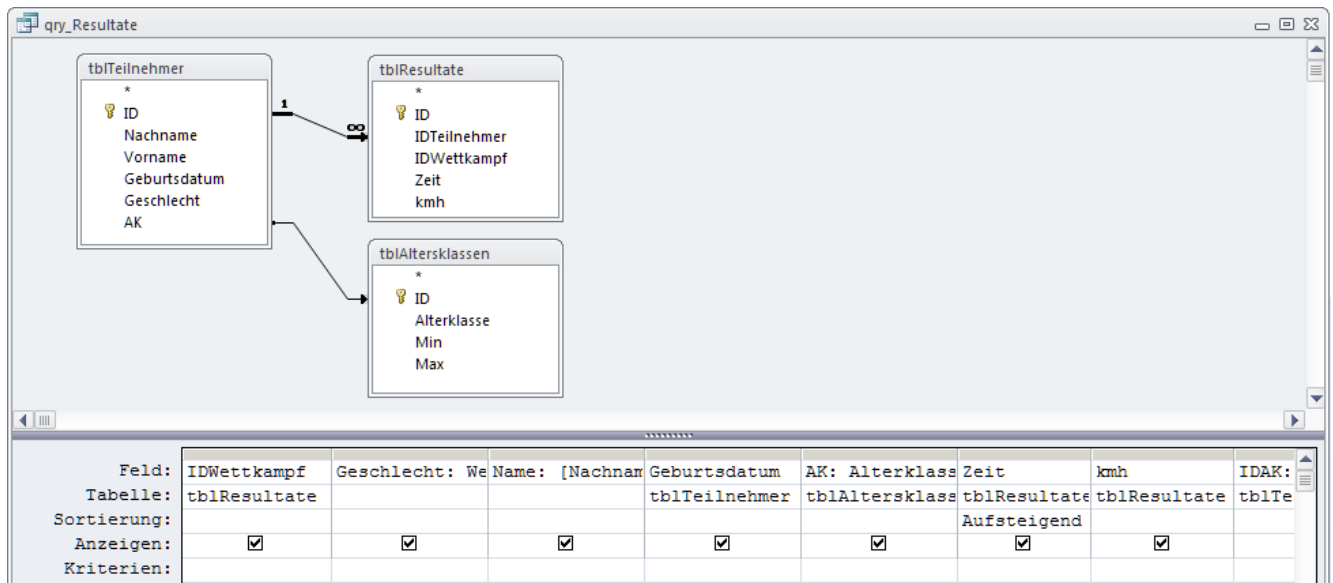


Bild 5: Diese Abfrage stellt die Datenherkunft des Unterberichts dar und kombiniert drei Tabellen zur Darstellung der Resultate

Die Felder der Datenherkunft **qry_Resultate** (siehe Bild 5) liegen nebeneinander im Detailbereich, so dass sich letztendlich eine Tabellendarstellung ergibt. Das Feld für die Geschwindigkeit (kmh) ist zwei Textboxen zugewiesen, wobei die linke den Wert anzeigt, die rechte einen korrelierenden Farbbalken. Für das Namensfeld ist die Eigenschaft **Vergrößerbar auf Ja** eingestellt, damit längere Teilnehmernamen nicht abgeschnitten werden, sondern das Textfeld nach unten ausdehnen. Hier zeigt sich bereits ein Vorteil des Unterberichts gegenüber einem Unterformular, denn dieses kann Steuerelemente ja nicht dynamisch skalieren.

Farbbalken über Bedingte Formatierung

Klicken Sie im Berichtsentwurf auf das rechte Textfeld und wählen aus dem Kontextmenü den Eintrag **Bedingte Formatierung...** Alternativ können Sie dafür die gleichnamige Schaltfläche im **Ribbon-Tab Format**

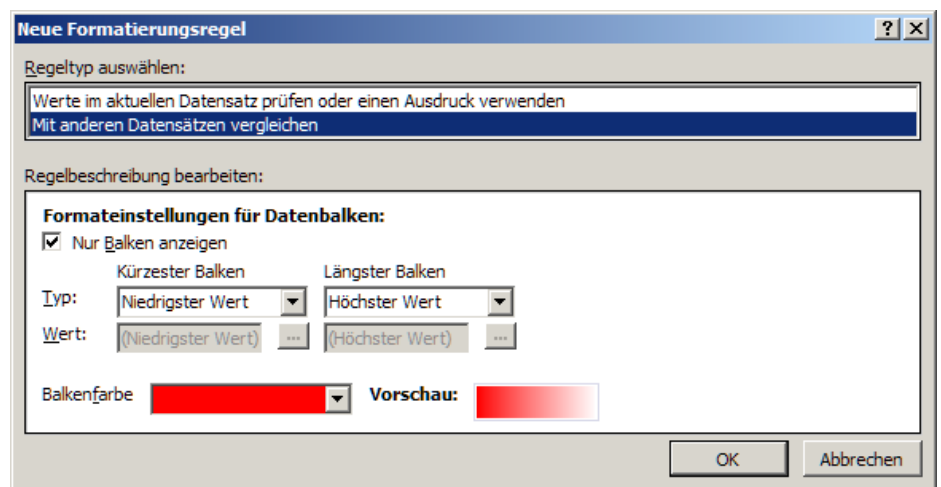


Bild 6: Der zweite Dialog zur Einstellung des Farbbalkens für die Feldformatierung

verwenden. Damit öffnet sich ein Dialogfenster zur Eingabe der Einstellungen für die Formatbedingungen, der zunächst leer ist. Sie legen deshalb eine neue Formatregel über den linken Button **Neue Regel an**, was einen untergeordneten Dialog hervorbringt (siehe Bild 6).

Entscheidend ist hier die Auswahl des **Regeltyps** oben. Mit **Werte im aktuellen Datensatz prüfen** haben Sie dieselben Einstellmöglichkeiten, wie auch früher schon in anderen Access-Versionen, und kön-

nen etwa Schriftfarbe oder Hintergrund des Textfelds in Abhängigkeit vom Wert oder von einem Vergleichsausdruck bestimmen. Mit dem Typ **Mit anderen Datensätzen vergleichen** aber ändert sich die Gestalt des Dialogs komplett. Man hätte diesen Typ auch gleich **Farbbalken** nennen können.

Aktivieren Sie **Nur Balken anzeigen**, so fehlt im Feld nachher die textuelle Wertangabe. Die Farbe des Balkens bestimmen Sie im entsprechenden Kombinationsfeld unten, die Vorschau verdeutlicht, wie das dann ungefähr aussehen wird. Grundsätzlich können Sie keinen Farbverlauf zwischen zwei Farben einstellen, denn links ist immer die gewählte Farbe, rechts der Verlauf nach Weiß. Denn die Vorschau lässt hier den Hintergrund des Textfelds außer Acht und rendert diesen nicht.

Tatsächlich handelt es sich rechts aber nicht um Weiß, sondern um den Alpha-Kanal des Balkens. Das bedeutet, dass hier der Hintergrund des Textfelds zunehmend durchscheint. Im Entwurf weiter oben ist zu sehen, dass der Hintergrund auf ein Schwarz eingestellt ist. Somit entsteht schließlich ein Verlauf von Rot nach Schwarz.

Ebenso könnten Sie auch andere Hintergründe verwenden und den Verlauf damit modifizieren. Sogar mehrere Balkenregeln parallel sind möglich, wodurch im Verein mit den Bedingungswerten weitere Gestaltungen entstehen können.

Wir haben für den Bedingungstyp im Beispiel die Angaben **Niedrigster Wert** und **Höchster Wert** gemacht. Damit ermittelt die Bedingung den kleinsten aller Geschwindigkeitswerte der Abfrage, welcher rein Rot wird, und den größten, der vollständig transparent wird. Zwischen diesen Polen bewegt dann sich der Verlauf.

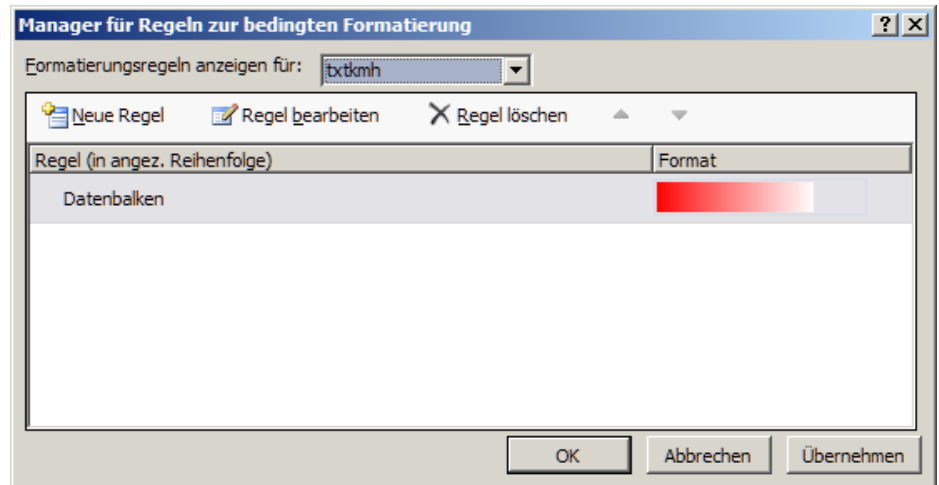


Bild 7: Der erste Dialog zur Einstellung der Bedingten Formatierung eines Textfelds

Komplizierter wird es bei der Auswahl von **Zahl** für den Typ. Hier könnten Sie in den dann aktivierten **Wert-Feldern** selbst Werte eintragen, oder sogar über Ausdrücke errechnen lassen, die den Bereich bestimmen. Wir belassen es jedoch bei der einfachsten Variante.

Nach dem Bestätigen mit **OK** schließt sich der Detaildialog und im Hauptdialog findet sich nun die neue Regel (Bild 7), die Sie übrigens über Doppelklick auch später noch verändern können. Außerdem können Sie nun auch weitere Regeln anlegen, wobei das nicht zwingend Farbbalken sein müssen, sondern ebenso die üblichen Bedingten Formatierungen für Schriftart und Hintergrund des Felds. Experimentieren Sie einfach etwas mit einigen Kombinationen herum, um die Gestaltungsmöglichkeiten auszuloten.

Bericht im Formular

Den nun fertiggestellten Bericht integrieren Sie in den Detailbereich des Formulars **frmErgebnisse-Report**, wie in Bild 8, indem Sie ein Steuerelement **Unterformular/-bericht** (Ribbon-Tab **Entwurf**) in ihm anlegen. Stellen Sie für die Eigenschaft **Verankern** das **Nach unten und quer dehnen** ein (Ribbon-Tab **Anordnen**), damit der Unterbericht möglichen Größenänderungen des Detailbereichs folgt. Als **Herkunftsobjekt** legen Sie **Bericht.rptErgebnisse** fest. Da das Formular selbst keine Datenherkunft aufweist, entfallen hier entsprechende Feldverknüpfungen.

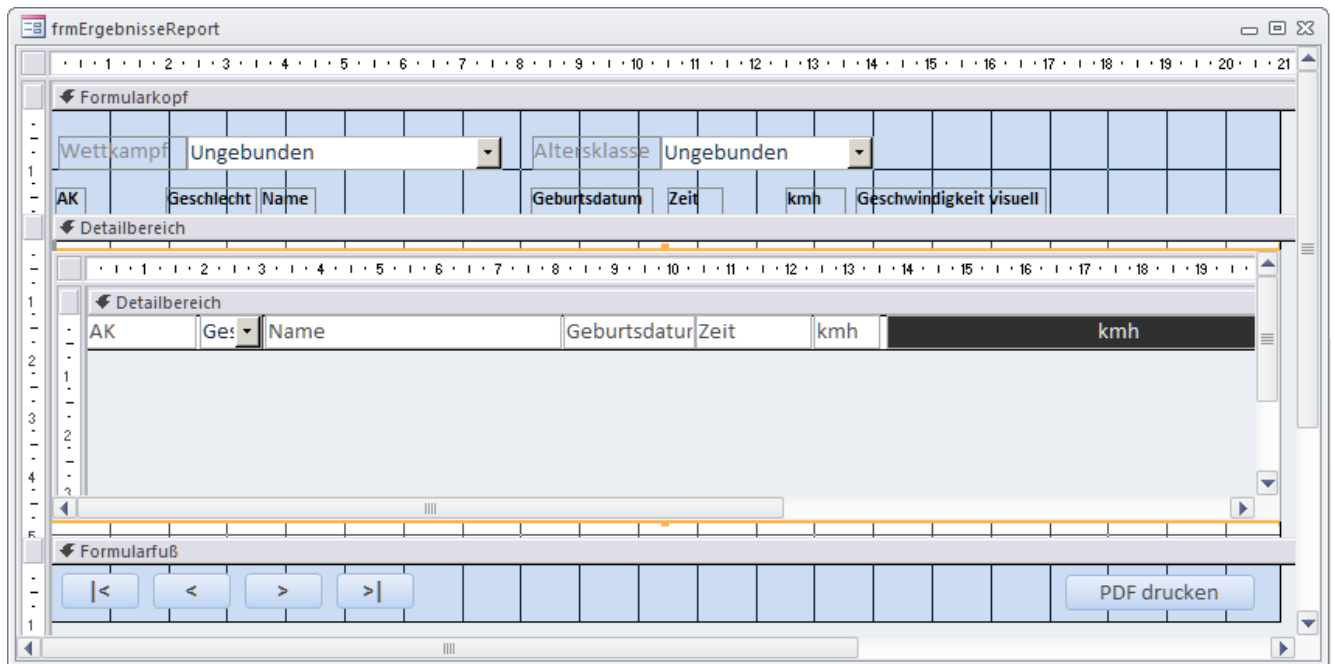


Bild 8: Im Entwurf von `frmErgebnisseReport` ist der Bericht als Steuerelement in den Detailbereich des Formulars integriert worden

Zum Filtern der Berichtsdaten dienen die beiden Kombinationsfelder im Formularkopf. Die Combo `cbWettkampf` holt ihre Einträge direkt aus der Tabelle `tblWettkampf`, die Combo `cbAltersklasse` aus einer UNION-Abfrage:

```
(SELECT 0 AS ID, "(Alle)" AS AK
FROM tblAltersklassen)
UNION (SELECT ID, Altersklasse AS AK
FROM tblAltersklassen)
```

Im Grunde kommen auch hier die Einträge aus der Nachschlagetabelle `tblAltersklassen`, werden jedoch mit dem Dummy-Eintrag `0` für `(Alle)` kombiniert. Dieser Eintrag mit der `ID 0` gibt später die Möglichkeit, eine Filterung nach Altersklassen auszuschalten.

Im Fußbereich des Formulars befinden sich vier Schaltflächen zur Navigation durch

die Datensätze des Unterberichts – auch das geht halbwegs in der Berichtsansicht! – und eine zur Ausgabe desselben in eine PDF-Datei.

Filtern des Unterberichts zur Laufzeit

Wählen Sie aus der Combobox `cbWettkampf` etwa den Eintrag `Promis` aus, so löst dies ihr Ereignis `Nach Aktualisieren (After_Update)` aus. Dasselbe geschieht bei Auswahl einer `Altersklasse (cbAK)`. Letztere Ereignisprozedur verweist lediglich auf die andere Ereignisprozedur, welche schließlich beide Filterungen zusammenfasst:

```
Private Sub cbWettkampf_AfterUpdate()
    If cbAK.Value = 0 Then
        Me!SubRpt.Report.Filter = "[IDWettkampf]=" & Me!cbWettkampf.Value
    Else
        Me!SubRpt.Report.Filter = "[IDWettkampf]=" & Me!cbWettkampf.Value & _
            "AND [IDAK]=" & cbAK.Value
    End If
    sRptFilter = Me!SubRpt.Report.Filter
    Me!SubRpt.Report.FilterOn = True
End Sub
```

Listing 1: Textverschlüsselung über eine String-Umkehr

```
Private Sub cbAK_AfterUpdate()  
    cbWettkampf_AfterUpdate  
End Sub
```

Die eigentliche Routine steht in Listing 1. Hier entscheidet eine Bedingung zunächst, ob in der Altersklassen-Combo der Dummy-Eintrag 0 gewählt wurde. Dann nämlich bildet sich der **Filter**-String etwas anders und begrenzt die Datensätze auf alle, die im Feld **IDWettkampf** den in der Combo gewählten Wert aufweisen. Andernfalls wird auch das Feld **IDAK** für die gewählte Altersklasse berücksichtigt und die Bedingung im String per **AND** verknüpft.

Dieser Filter-String wird dem Unterbericht verarbeitet, der als Steuerelement im Formular die Bezeichnung **SubRpt** trägt. Wichtig ist, dass Sie dabei nicht das Steuerelement selbst ansprechen, sondern dessen Herkunft mit dem Objektverweis **Report**:

```
Me!SubRpt.Report
```

Der Filter-String wird gültig, sobald die Eigenschaft **FilterOn** auf **True** gesetzt wird. Augenblicklich setzt die Neuabfrage der Datensätze ein und die Liste im Formular aktualisiert sich.

Damit unterscheidet sich die Filterung in keiner Weise von der, die Sie auch für ein Unterformular vorgenommen hätten.

Übrigens ist für die Wettkampf-Combo im Entwurf der **Standardwert 1** voreingestellt, was dem Wettbewerb **55km-Frauen** entspricht, und für die Altersklassen-Combo der Wert **0 (Alle)**.

Damit der Unterbericht dann beim Laden auch die passenden Daten anzeigt, erfolgt in der Ereignisprozedur **Beim Laden (Form_Load)** bereits die erste Filterung:

```
Private Sub Form_Load()  
    Me!SubRpt.Report.Filter = "[IDWettkampf]=1"  
    Me!SubRpt.Report.FilterOn = True  
End Sub
```

Im Listing wird der Filterausdruck zusätzlich der Variablen **sRptFilter** zugewiesen. Sie nimmt damit den gegenwärtigen Filter auf. Sie steht dem VBA-Projekt immer zur Verfügung, da sie im Modul **mdlReport-helper** als **Public** deklariert ist. Wofür wir deren Inhalt noch benötigt wird, zeigt sich später.

An sich ist die Navigation durch die Datensätze eines Berichts wie in einem Datenblatt eine unnötige Angelegenheit. Zur Demonstration, dass jedoch auch dies in einem Bericht in Berichtsansicht möglich ist, sind die vier Schaltflächen im Formular eingebaut. Der Button zum Anspringen des ersten Datensatzes (**cmdFirst**) löst etwa diese Ereignisprozedur aus:

```
Private Sub cmdFirst_Click()  
    Me!SubRpt.SetFocus  
    DoCmd.GoToRecord , , acFirst  
    Me!SubRpt!AK.SetFocus  
End Sub
```

Die entscheidende Anweisung ist hier das **DoCmd.GotoRecord** mit dem Parameter **acFirst**. Sie wirkt sich auf das in der Oberfläche aktive Objekt aus.

Das soll ja der Unterbericht sein, weshalb der Fokus zuerst auf das ihn enthaltende Steuerelement **SubRpt** gelegt wird. Außerdem legt die Routine noch den Fokus auf das Textfeld **AK** des Berichts. Auf die gleiche Weise gelangt man zum letzten Datensatz des Berichts:

```
Private Sub cmdLast_Click()  
    Me!SubRpt.SetFocus  
    DoCmd.GoToRecord , , acLast  
    Me!SubRpt!AK.SetFocus  
End Sub
```

Die Darstellung der Navigationsroutinen für den nächsten oder vorigen Datensatz sparen wir uns hier – sie geschieht analog.

Was allerdings nicht gelang, ist das Markieren der dann aktiven Datensatzzeile. Dafür kann man etwa diese Anweisung verwenden:

RunCommand acCmdSelectRecord

Das Markieren erfolgt damit zwar tatsächlich, doch anschließend springt der Bericht eigenartigerweise immer auf den ersten Datensatz...

Es lässt sich also zusammenfassen, dass sich ein Bericht in der **Berichtsansicht** weitgehend so verhält, wie ein Formular. Der Vorteil gegenüber einem Unterformular liegt in erweiterten Formatierungsmöglichkeiten und den Gruppierungsfunktionen, die wir bisher noch gar nicht ansprachen. Natürlich kann ein Bericht in dieser Ansicht auch wie üblich in Ebenen gruppiert werden, was in Formularen nur über verschachtelte Unterformulare zu realisieren wäre. Dieses Feature ließen wir in der Beispieldatenbank aber außen vor.

Berichtsausgabe in PDF-Datei

Neben dem Drucken eines Berichts ist seine zweitwichtigste Anwendung sicher die Ausgabe in ein PDF-Dokument. Das dürfte zumindest seit Access 2010 der Fall sein, als diese Ausgabemöglichkeit fest in Office eingebaut wurde.

Natürlich können Sie auch auf einen installierten PDF-Drucker ausgeben, doch diesen setzen wir zunächst nicht voraus. Formulare, Datenblätter und Bericht kann Access mit Bordmitteln als PDF rendern. Dafür gibt es genau eine Methode des **DoCmd**-Objekts. Beispiel:

```
DoCmd.OutputTo acOutputReport, " _
    "rptErgebnisse2", _
    acFormatPDF, _
    CurrentProject.Path & "\results.pdf", _
    True
```

OutputTo ist eine recht variable Anweisung, die Access-Objekte in allerlei Formaten ausgeben kann. Mit dem Steuerparameter **acFormatPDF** kommt es zum PDF-Export. Dass das gewünschte Objekt ein Bericht ist, sagt **acOutputReport** aus.

Dessen Name (**rptErgebnisse2**) folgt als nächster Parameter. Für das PDF braucht es zusätzlich einen Dateinamen, der sich hier aus dem Pfad der Datenbank plus der Bezeichnung **results.pdf** zusammensetzt.

Schließlich können Sie als letzten Parameter optional angeben, ob das erzeugte Dokument sogleich im zuständigen Viewer angezeigt werden soll, was mit dem Wert **True** festgelegt wird. Mit **False** unterbleibt das und das Dokument wird im Stillen angelegt. Allerdings zeigt dann dennoch ein Dialog den seitenweisen Fortschritt des Renderns an.

Da die Methode die dezidierte Angabe eines Objekt-namens erfordert und nicht etwa, wie viele andere **DoCmd**-Anweisungen, auf das gerade aktive Objekt wirkt, kann der Unterbericht unseres Beispielformulars leider nicht auf direktem Wege in ein PDF-Dokument ausgegeben werden. Das klappt nur und ausschließlich über den Umweg eines anderen Berichts.

Kopieren Sie also den Bericht **rptErgebnisse** und benennen ihn mit **rptErgebnisse2**. Löschen Sie den gesamten VBA-Code des Berichts. Erweitern Sie die Gestaltung, wie in Bild 10 vorgestellt.

Neben den eigentlichen Daten im Detailbereich sind hier noch Spaltenköpfe im

AK	Geschlecht	Name	Geburtsdatum	Zeit	kmh	Geschwindigkeit visuell
Sen III	w	Sechelmann, Werner	07.04.1960	03:24:48	43,944	
Sen III	w	Terrahé, Michael	20.06.1962	03:27:02	43,471	
Sen III	w	Oldenburg, Andreas	24.06.1959	03:27:59	43,273	
Sen III	w	Henke, Heinrich	29.11.1959	03:28:53	43,086	
Sen III	w	Schüppel, Ekkehard	30.11.1961	03:32:27	42,364	
Sen III	w	Plöhn, Hanns-Joachim	12.12.1961	03:33:15	42,204	
Sen III	w	Keppeler, Dominik	08.03.1965	03:37:58	41,291	
Sen III	w	Wolf, Michael	14.07.1956	03:43:38	40,245	
Sen III	w	Markowsky, Bernd	30.04.1965	03:46:10	39,793	
Sen III	w	Thomas, Walter	04.12.1957	03:56:04	38,126	
Sen III	w	Dechert, Manfred	20.08.1963	03:57:11	37,946	
Sen III	w	Bünger, Michael	29.08.1962	03:59:44	37,541	
Sen III	w	Will, Klaus-Peter	14.12.1964	04:02:51	37,060	
Sen III	w	Hoffmann, Georg	14.05.1957	04:04:33	36,801	
Sen III	w	Willems, Willi	14.05.1965	04:07:15	36,402	
Sen III	w	Kratsch, Rudolf	20.07.1962	04:09:25	36,085	
Sen III	w	Walsemann, Michael	05.01.1956	04:10:03	35,992	

Bild 9: So präsentiert sich das exportierte PDF-Dokument mit den Resultaten

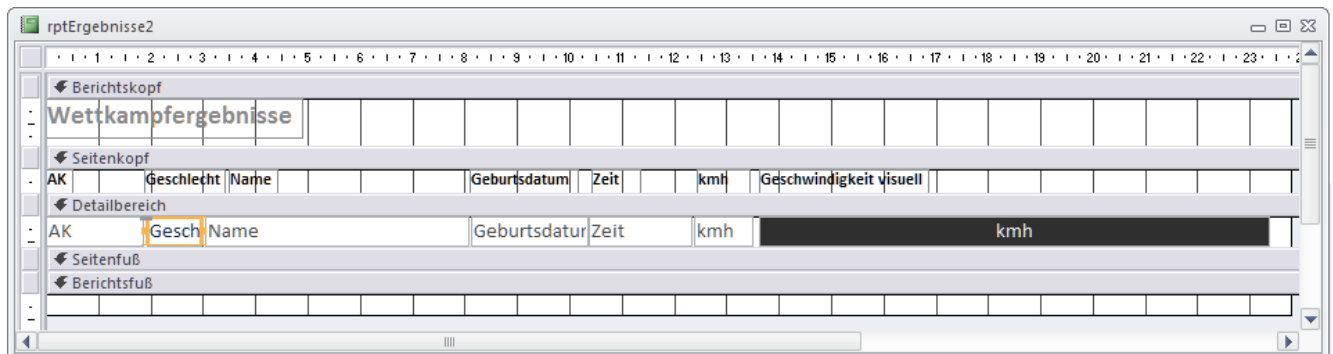


Bild 10: Der Bericht rptErgebnisse2 dient dem Ausdruck der gefilterten Resultate und dem Export in das PDF-Format

Seitenkopf eingefügt und zusätzlich ein Titel im Berichtskopf. All dies sind Labels.

Einen statischen Bericht filtern

Dieser Bericht wird im Beispielformular durch Klick auf die Schaltfläche **PDF drucken** und die erwähnte **OutputTo**-Methode aufgerufen. Dabei würden allerdings sämtliche Ergebnisse aller Wettkämpfe ungefiltert in das Dokument geraten. Um das zu verhindern, setzt die Ereignisprozedur **Beim Öffnen (Report_Open)** des Berichts gleich einen Filter:

```
Private Sub Report_Open(Cancel As Integer)
    Me.Filter = sRptFilter
    Me.FilterOn = True
End Sub
```

Nun wissen Sie, wofür der Filter-String des Unterberichts des Formulars zuvor in der globalen Variablen **sRptFilter** abgelegt wurde. Er kommt hier abermals

zum Einsatz. Das **Beim Öffnen**-Ereignis ist übrigens der einzige Ort, an dem Sie einen Bericht in Seitenansicht – diese impliziert **OutputTo** nämlich – filtern können. Zu späteren Zeitpunkten gelingt dies absolut nicht mehr.

Alternativ hätte man die dem Bericht zugrundeliegende Abfrage modifizieren und für ihre Filterparameter globale Funktionen einsetzen können, die ebenfalls über das Formular gesetzt würden. Tatsächlich wäre diese Lösung performanter, da der Bericht beim Öffnen so nur eine Teilmenge der Datensätze entgegennehmen müsste und das Filtern entfiel.

Dem steht allerdings ein höherer Aufwand mit mehr Objektabhängigkeiten entgegen.

Wie auch immer, das fertige PDF-Dokument zeigt sich im Viewer etwa so, wie in Bild 9. Auch die Farbbalken werden anstandslos gerendert.