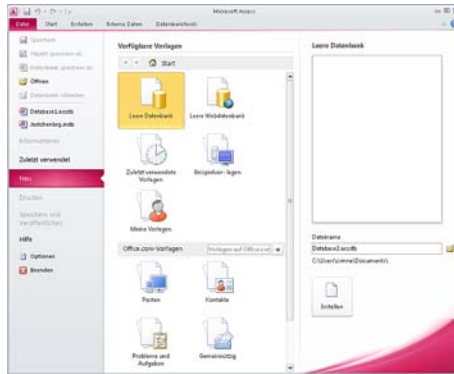


**Erstellen einer
archäologischen Fundplatzdatenbank
mit
MS Access 2010**



Erstellt für eine Übung im Wintersemester 20011/2012

von Christoph Rinne

aktualisiert: November 2011

Der folgende Text ist *a priori* als Skript für die Übung „Spätneolithische Siedlungen in Mitteldeutschland“ konzipiert und greift somit nicht alle vor- oder auch nachbereitenden Unterrichtseinheiten und auch begleitenden Erläuterungen auf. Zudem setzt diese Einführung gewisse Grundkenntnisse voraus, für die auf gute allgemeine Einführungen verwiesen wird. Erwähnt seien hier die kostengünstigen Handbücher des RRZN (<http://www.rrzn.uni-hannover.de/buecher.html>). Hinzu kommen zahlreiche Internetseiten, von denen hier aus der eigenen Erfahrung als Lernender folgende erwähnt sei: <http://www.donkarl.com/>

- ⚠ Bei dem nachfolgenden Skript handelt sich nicht um eine Einführung in die Gestaltung relationaler Datenbanken. Die wichtige Normalisierung von Datenbeständen und das Schlüsselkonzept werden nicht (!) behandelt.

Es wird eine pragmatische Lösung mit einer basalen relationalen Datenstruktur für Altdaten aus der Literatur erstellt. Für ein sehr breit aufgestelltes relationales Datenbankschema sei auf das Modell des English Heritage's Centre for Archaeology at Fort Cumberland, bei Portsmouth verwiesen (http://cidoc.ics.forth.gr/docs/AppendixA_DiagramV9.pdf). Eine deutlich schlankere Form wird in der Literaturdatenbank Zotero realisiert (knappe Erläuterung siehe z. B. Rinne, [Einführung Zotero](#))

Zum Layout seien noch folgende Hinweise vorangestellt.

Menüpfade werden wie folgt dargestellt: Datei → speichern unter

Tastatureingaben werden wie folgt dargestellt: <Alt>+<F1>

Schalter auf einem Formular werden wie folgt dargestellt: [Weiter >]

Eingaben in einem Feld werden in Anführungszeichen gesetzt, die nicht zu übernehmen sind.

Code wird in der Schrifttype Courier dargestellt: `SELECT * FROM t_befunde;`

1 Starten von MS Access

Wenn Sie MS Access® (im Folgenden nur Access) starten, wird Ihnen ein Startfenster mit vielen Optionen präsentiert. Die Vorauswahl erstellt Ihnen 1. automatisch eine Datenbank in Ihrem Nutzerverzeichnis (\Benutzer\Eigene Dokumente\Database1.accdb) und in dieser wiederum eine Tabelle (Tabelle1). Anders als in Word und anderen Office Programmen werden Sie beim Speichern nicht automatisch nach Name und Ort gefragt! Den Namen der Datenbank und den Speicherort können Sie im rechten unteren Viertel des Fensters angeben.

Der Verfasser empfiehlt deshalb folgendes: wechseln Sie in den Ordner, an dem die Datenbank liegen soll und wählen Sie im Kontextmenü – rechte Mausklick innerhalb eines leeren Fensterbereichs – die Option: Neu → Microsoft Access Datenbank. Geben Sie der Datenbank einen Namen und starten Sie diese direkt mit einem Mausklick.

1.1 Das Programm- oder Steuerungsfenster

Beim ersten Öffnen einer DB erfolgt ggf. eine Sicherheitswarnung. Da es sich um Ihre eigene Datenbank handelt, wählen Sie bitte [Inhalte aktivieren]. Nähere Informationen zum Thema Makrosicherheit in Access finden Sie bei der Suche nach dem Stichwort „Sandbox Mode“. [Nicht

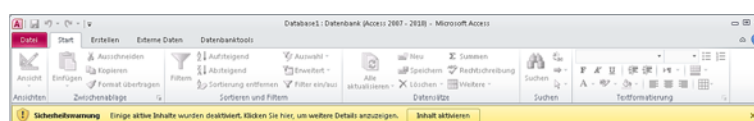


Abb. 1 Sicherheitswarnung. Bei Ihren eigenen Datenbanken können Sie Inhalte wohl bedenkenlos aktivieren.

aktivieren] deaktiviert auch basale Funktionen von MS Access.

In dem Programmfenster steht oben der Datenbankname, darunter folgt das Ribbon, links finden Sie ein „Inhaltsverzeichnis“ der Objekte Ihrer DB, rechts eine jeweils wechselnde Ansicht zum gewählten Objekt im Inhaltsverzeichnis und der gewählten Aktivität.

- Für Umsteiger: einzelne, überlappende Fenster erhalten Sie über: Datei → Optionen, Register Aktuelle Datenbank und Option Überlappende Fenster auswählen.
- Für Umsteiger: Ausgeblendete Objekte und Systemobjekte anzeigen. Rechtsklick auf die Kopfzeile des „Inhaltsverzeichnis“ (z.B. „Alle Access-Objekte“) → Navigationsoptionen → Anzeigoptionen

Das Inhaltsverzeichnis links lässt sich unterschiedlich gestalten, wählen Sie die jeweilige Option aus dem Drop-Down-Menü am oberen Rand. Für den Anfang kann hier sicher „Alle Access-Objekte“ ausgewählt werden, später sollten Sie hier mit Gruppenfiltern oder der benutzerdefinierten Einstellung arbeiten. Entsprechend dem Inhaltsverzeichnis kann Ihre Datenbank *prima vista* Tabellen, Abfragen, Formulare und Berichte enthalten.

- Tabellen: enthalten die Daten, Sie dienen nicht(!) der Datenerhebung
- Abfragen: bieten unterschiedliche Sichten auf die Daten, z.B. gefiltert, oder erlauben Änderungen und Ergänzungen an den Daten
- Formulare: dienen der Interaktion zwischen Nutzer und DB, z.B. der Eingabe, sie dienen nicht(!) dem Ausdruck
- Berichte: dienen dem Druck der Informationen

2 Tabellen

2.1 Erstellen einer Tabelle

Erstellen → Tabellenentwurf erlaubt Ihnen Entscheidungen zur Gestaltung.

In dem sich öffnenden Fenster sehen Sie eine Tabelle für die Feldnamen, den Feldtyp und eine Beschreibung für jede Spalte (Attribut) der späteren Tabelle. Alle Spalten sind wichtig, unterschätzen Sie nicht die Bedeutung der Beschreibung: a) Dieser Text wird in der Statuszeile des Accessfensters ausgegeben und ist somit eine wichtige Information für den Nutzer, b) zur Unterstützung Ihres persönlichen Gedächtnis.

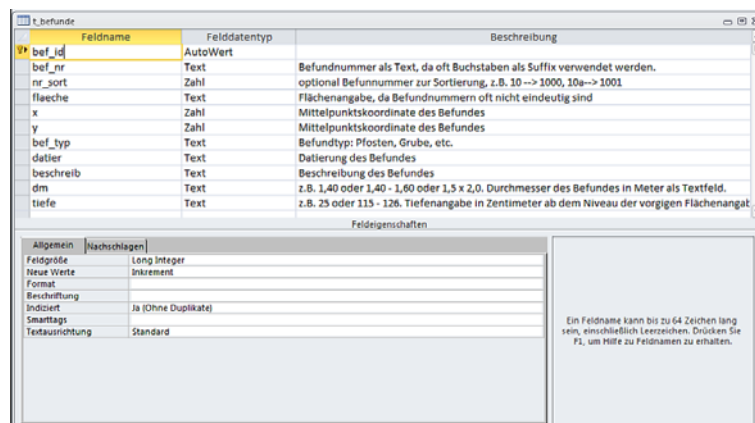



Abb. 2 Mögliche Struktur einer Befundtabelle.


 Beherzigen Sie bei den Feldnamen folgendes:

- seien Sie stringent, verwenden Sie keine Sonderzeichen, Umlaute und das Leerzeichen, fassen Sie sich kurz ohne kryptisch zu werden, Feldnamen beginnen nicht mit einer Ziffer, Begriffe von Funktionen sind zu meiden (z. B. Summe und Anzahl)
- gleiches gilt für Tabellennamen.

2.1.1 Befundtabelle t_befunde

Eine Tabelle für die Befunde könnte wie in Abb. 2 gestaltet werden. Überdenken Sie folgende Problematik zum Schlüsselkonzept relationaler Datenbanken und der Realität auf Grabungen und Altgrabungen: Ist die Befundnummer wirklich ein Schlüsselfeld oder ergibt sich der Schlüssel erst in Kombination mit weiteren Feldern wie Fläche, Jahr der Grabung, Fundplatz etc.?. Wählen Sie wie in Abb. 2 eine automatisch generierte ID, ist diese Problem leicht behoben. Für die Relation zu transitiv abhängigen Tabellen (z. B. t_funde) reicht aber nicht die Befundnummer – die in der Grabungsdokumentation in der Regel das Schlüsselfeld ist – sondern es muss diese „nichtssagende“ neue ID angegeben werden.

Beachten Sie die Optionen für jedes Attribut in der unteren Fensterhälfte. Neben dem Register „Allgemein“ finden Sie auch „Nachschlagen“. Tragen Sie für das Feld bef_typ hier folgendes ein: Steuerelementanzeige: Kombinationsfeld, Herkunftstyp: Tabelle/Abfrage, Datensatzherkunft: `SELECT t_befunde.bef_typ FROM t_befunde GROUP BY t_befunde.bef_typ ORDER BY Count(t_befunde.bef_id) DESC;`, Gebundene Spalte 1, Spaltenanzahl 1. Ohne in's Detail zu gehen, die vorige SQL-Syntax fragt alle vorhandenen Einträge im Attribut bef_typ ab und sortiert sie absteigend nach deren Häufigkeit. Effekt: a) der Nutzer erhält eine Liste der bisherigen Einträge zur Auswahl, b) die Liste wird automatisch um neue Einträge ergänzt, c) bekannte Begriffe werden automatisch ergänzt. Ziel: ein möglichst stringenter Datenbestand.

 Es gibt zwei diametral entgegengesetzte Strategien bei der Gestaltung von Tabellen in Access.

- Die Tabelle beinhaltet nur die Daten; Formate, Feldbeschriftung, Gültigkeitsregeln und auch Nachschlagedefinition für Wertecodes haben hier nichts verloren. Begründung: in der Tabellenansicht entspricht die Spaltenüberschrift nicht dem Feldnamen, Abfragen müssen für den hinterlegten Code und nicht für den angezeigte Wert formuliert werden. Dies führt zwangsläufig zu Irritationen beim Nutzer.
- Definiere ich an dieser Stelle Format, Beschriftung und Nachschlagefeld, wird dies für Formulare und Berichte zu dieser Tabelle automatisch übernommen. Ich spare Arbeit und vermeide unbewusste Änderungen.

2.1.2 Fundtabelle t_funde

Erstellen Sie eine Tabelle mit folgenden Feldern: fund_id (Long Integer); bef_nr (Text); abc (Text); menge (Long Integer) gefaess (Text); erhaltung (Text); datier (Text); verzier (Text); randdm (Double); hoehe (Double); bodendm (Double); tafel (Text).

2.2 Importieren und Verknüpfen von Daten

Externe Daten können importiert, also in die Datenbank kopiert oder auch verknüpft werden. Letzteres bietet den Vorteil, stets auf den aktuellen Datenbestand einer externen Quelle zuzugreifen und ggf. auch editieren zu können, eine Strukturveränderung ist aber nicht möglich. Letzteres bietet vollständige Souveränität.

Wählen Sie im Ribbon den Reiter „Externe Daten“ und dort „Excel“. Wählen Sie die „Liste_Pfosten.xls“ aus und belassen Sie die Option „Importieren Sie ...“ markiert. Verwenden Sie das erste Arbeitsblatt und klicken Sie auf [Weiter >]. Die erste Zeile enthält die Feldnamen, die Fehlermeldung ignorieren Sie, da sie sich erkennbar auf die leeren Spalten rechts bezieht. Entscheidend ist das Fenster, in dem Sie die Datentypen der Felder bestimmen. Für die Felder „dm“ und „t“ ändern Sie diesen Wert bitte von „Double“ auf „Text“.

Wenn Sie Tabellen vorübergehend importieren und anschließend löschen sollten Sie die Datenbank reorganisieren und komprimieren. Dies kann die Datenbank beträchtlich schrumpfen.
Datenbanktools → Datenbank komprimieren ...

- Beim Import werden Excel-Tabellen gerne direkt, ohne Angaben zur Spezifikation importiert, das führt regelmäßig zu Fehlern. In unserem Fall wäre das bei den Feldern „dm“ und „tiefe“ geschehen. Der Import als csv- oder tab-Datei schafft Abhilfe.
- Spezifikationen werden in der Datenbank gespeichert und sind dann abrufbar.

3 Abfragen

Abfragen stellen primär unterschiedliche Sichten auf die Daten zusammen (Auswahlabfragen), weitere Abfragen überführen Daten in neue Tabellen, fügen Daten an bestehende Tabellen an oder löschen diese und aktualisieren Werte innerhalb eines Datenbestandes. Kreuztabelle erstellt eine Auswertungsmatrix, z. B. Anzahl der Werte in gefaesse je bef_nr. Abfragen enthalten *per se* keine Daten sondern nur eine Anweisung (SQL).

3.1 Anfügeabfrage

Die folgende Abfrage soll die Daten aus der importierten Tabelle an die vorhandene Tabelle t_befunde anfügen. Wählen Sie im Ribbon „Erstellen“ → „Abfrageentwurf“. Zuerst müssen Sie eine Tabelle für die Datenherkunft wählen, in diesem Fall die soeben importierte Excel Liste. Klicken Sie im Ribbon auf das Icon der Anfügeabfrage („Anfügen“) und geben Sie die tabelle „t_befunde“ der aktuellen Datenbank an. Felder mit identischen Feldnamen werden automatisch zugewiesen

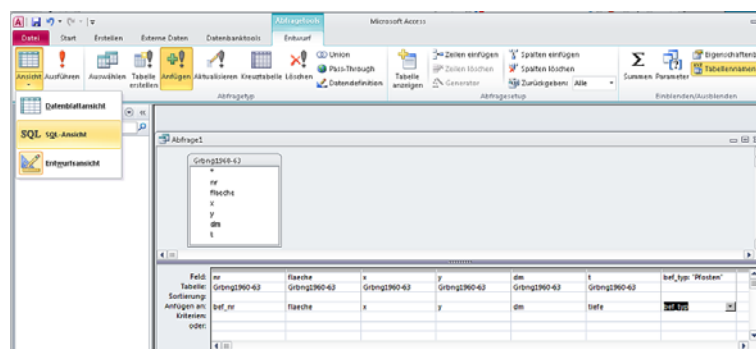


Abb. 3 Anfügeabfrage (QBE) im Access.

(Kontrolle!), der Rest muss von Hand zugewiesen werden. Da die Tabelle nur Pfoften enthält, fügen Sie in der ersten freien Spalte in der ersten Zeile bitte folgenden Text ein: „ bef_typ: "Pfoften" „ in der Zeile Anfügen an: wählen Sie „bef_typ“. Alle Einträge erhalten nun als Typ den Text „Pfoften“.

Dies Art der Gestaltung von Abfragen ist sehr intuitiv und wird bei Access als „Query By Example“ (QBE) bezeichnet. Werfen Sie dennoch einen Blick auf die SQL-Anweisung im Hintergrund: Ansicht → SQL-Ansicht. Der Befehl ist klar in drei Abschnitte gegliedert und durchaus verständlich.

```
INSERT INTO t_befunde ( bef_nr, flaeche, x, y, dm, tiefe, bef_typ )
SELECT [Grbng1960-63].nr, [Grbng1960-63].flaeche, [Grbng1960-63].x,
[Grbng1960-63].y, [Grbng1960-63].dm, [Grbng1960-63].t, "Pfoften" AS bef_typ
FROM [Grbng1960-63];
```

Prüfen Sie das Ergebnis in der Tabelle t_befunde, schließen Sie diese Abfrage und speichern Sie diese nicht, so viel Hirn steckt da nicht drin. Die zuvor importierte Excel Liste der Pfoften ist nun überflüssig und kann gelöscht werde (bei Gelegenheit Datenbanktools → Datenbank komprimieren ...).

3.2 Aktualisierungsabfrage

Daten – zumal „unsaubere“ Altdaten – müssen oft modifiziert werden. Als Beispiel dient hier das Feld bef_sort, das für die Sortierung erstellt wurde . Die nachfolgende Syntax zeigt einige Möglichkeiten von SQL und den in MS Access zur Verfügung stehenden Funktionen auf.

Erstellen Sie eine neue Abfrage über die Tabelle t_befunde. Überführen Sie das Attribut bef_nr mit Doppelklick in die Abfrage. In die erste Zeile der ersten freien Spalte überführen Sie bitte folgende Anweisung:

```
d1: Wenn(Rechts([bef_nr];1)="a";Links([bef_nr];Länge([bef_nr])-1)*100+1;[bef_nr]*100)
```

Führen Sie erstmal die Abfrage aus und betrachten Sie das Ergebnis. Nun zurück zur Syntax. „d1“ ist ein Dummy für einen Feldnamen der links vom Doppelpunkt stehen muss. Nach dem Doppelpunkt folgt eine Bedingung mit folgenden Elementen: wenn(Bedingung; tu dies; sonst dies).

Innerhalb dieser Therme werden weitere Funktionen verwendet, dies sind:

- rechts(feld;länge) – liefert die Zeichenkette der angegebenen Länge vom rechten Rand des Feldes
- links(feld;länge) - liefert die Zeichenkette der angegebenen Länge vom linken Rand des Feldes. Anmerkung: Teil(feld;start;länge) ist somit vermutlich verständlich.
- länge(feld) – liefert die Anzahl der Zeichen in einem Feld

Der Rest der Syntax besteht aus einfacher Addition und Multiplikation.

Kopieren Sie die gesamte bedingte Anweisung (wenn(...)) in die Zwischenablage, wechseln Sie zu einer Aktualisierungsabfrage für die Tabelle t_befunde, wählen Sie das Feld nr_sort aus und kopieren Sie die soeben erstellte Syntax in die Zeile „Aktualisieren:“ Führen Sie diese Abfrage aus.

Diese Abfrage ist recht primitiv, denn sie funktioniert nur für „a“.

4 Formulare

Formulare dienen der Interaktion mit dem Nutzer der Datenbank. Formulare können also ausschließlich Informationen bieten, Informationen aufnehmen, beides zugleich oder für die Navigation verwendet werden. In Formularen können Dateneingaben kontrolliert, an abhängige Datensätze weitergegeben und vor allem strukturiert in ansprechender Form dem Nutzer präsentiert werden. Formulare dienen nicht dem Ausdruck.

- 📘 Formulare ermöglichen eine kontrollierte und visuell aufbereitete Kommunikation, damit tragen Formulare wesentlich zur Datenqualität und -integrität bei.

4.1 Formularassistent

Starten Sie den Formularassistent: Erstellen → Formular-Assistent. Im folgenden Fenster können Sie eine Tabelle oder Abfrage als Datenquelle wählen, nachfolgend die darzustellenden Felder.

Anmerkung: Bei älteren Access-Versionen stand nicht „Tabelle:“ vor den Tabellen weshalb es u.a. üblich ist, für Tabellen, Abfragen, Formulare und Berichte ein Präfix zu verwenden (tbl, qry, frm, rpt, vgl. „Leszynski naming convention“). Wählen Sie alle Attribute aus [>>] und dann [Weiter >].

Im folgenden Fenster können Sie die Gliederung des Formulars bestimmen. Sehen Sie sich einmalig die einzelnen Vorlagen an und entscheiden Sie sich für „Einspaltig“. Der visuelle Unterschied zwischen „Tabellarisch“ und „Datenblatt“ ist hier minimal, ersteres sieht später wie ein Formular, letzteres wie eine Tabelle aus. Mit der Tabellenansicht können Sie Nutzern z. B. suggerieren, direkt in den Daten zu arbeiten. Im letzten Fenster vergeben Sie bitte einen sinnvollen Namen. Sinnvoll bedeutet z. B. „f_“ und je nach Anzahl der Formulare in der DB eine weitere Gliederungsinformation, z. B. „ed_“ für edit, insgesamt also z. B. „f_ed_befunde“. Belassen Sie die Option „Formular öffnen ...“ wählen Sie [Fertig stellen].

- 📘 Sinnvolle und gegliederte Formularnamen erleichtern dem Gestalter die Arbeit und sind für den Nutzer unabdingbar solange keine Navigation vorhanden ist. Beherrigen Sie die Hinweise zu Feldnamen und Tabellennamen.

Das Ergebnis des Assistenten ist nicht wirklich überzeugend, deshalb wechseln Sie in die

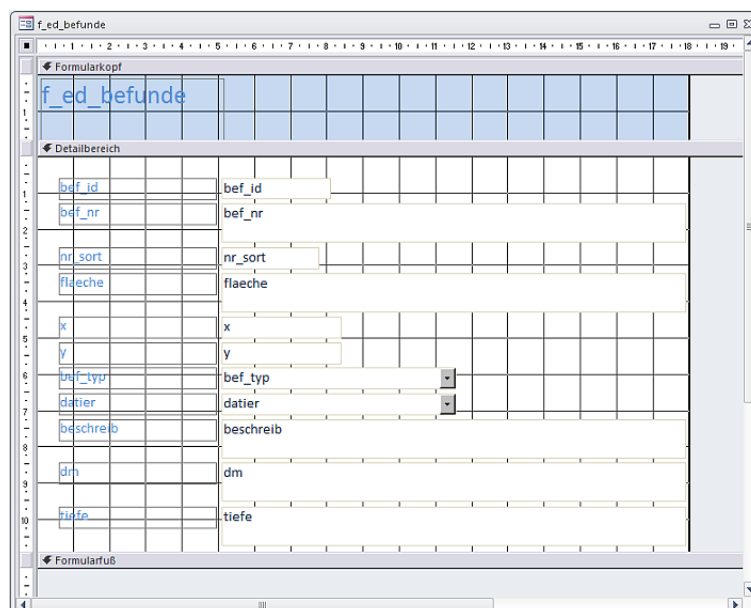


Abb. 4 Entwurfsansicht des Formulars

Entwurfsansicht Start → Ansicht → Entwurfsansicht (Abb. 4).

4.2 Formularentwurf

Die einzelnen Elemente des Formularentwurfes sind: der Formulkopf, der Detailbereich mit Beschriftungs- und Datenfeldern und der Formularfuß, der vom unteren Rand des Balkens mit der Maus aufgezogen werden kann. Jedes Element vom Formular insgesamt über die Bereiche bis hin zum Steuerelement (z. B. Datenfeld) hat Eigenschaften.

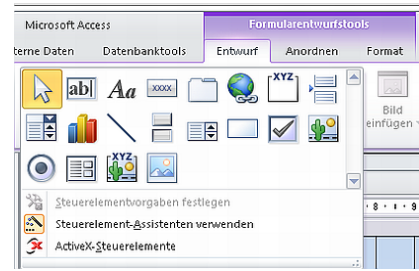


Abb. 5 Steuerelemente für ein Formular in der Entwurfsansicht

Eine Auflistung aller Felder (Attribute) erhalten Sie mit Formularentwurfstools → Entwurf → Vorhandene Felder hinzufügen. Daneben finden Sie den Schalter zum Einblenden des Eigenschaftenblatts, dieses passt sich an das jeweils markierte Steuerelement an oder Sie wählen das Steuerelement aus dem oberen Drop-Down-Menü des Eigenschaftenblattes aus. Aktivierreihenfolge bestimmt die Reihenfolge, in der die Felder des Formulars mit <TAB> durchlaufen werden.

- Ein gutes Formular ist übersichtlich, überfrachtet den Nutzer nicht mit Informationen, bietet die Option von hier aus zu übergeordneten oder abhängigen Daten zu wechseln, zu Filtern oder zu suchen und das Ergebnis zu drucken. Kurz: gute Formulare sind schwierig.

4.2.1 Steuerelemente eines Formulars

Eine Übersicht der Steuerelemente finden Sie im mittleren Bereich des Ribbon Formularentwurfstools → Entwurf → Steuerelement (Abb. 5). Nach dem Auswahlinstrument (Pfeil) folgen:

- Textfeld: ein Datenbankfeld, auch Zahlen,
- Bezeichnung: Feldbezeichnungen oder einfache Textfelder zur Information,
- Schaltflächen: für Makros und VBA-Anweisungen, z. B. Formular oder Bericht öffnen,
- Registerseteuerelement: erstellt Register auf dem Formular und gliedert so den Inhalt (gut), alle Daten auf allen Registern werden abgerufen, auch wenn der Nutzer die Daten nicht sieht oder auch nicht sehen will (schlecht)
- Hyperlink
- Optionsgruppe: gliedert das Formular visuell und definiert abhängige Auswahloptionen, sog. *radio buttons*
- Seitenumbruch: Für die Gestaltung von Berichten
- Kombinationsfeld: Drop-Down-Feld zum Auswählen von Vorgaben
- Diagramm: Auswertung von Daten
- Linien: Grafikelement für optische Gliederung, lässt sich der Formulkopf oder Fuß nicht „zuschieben“, liegt oft eine Linie im Weg
- Umschaltfläche: im Grunde nur an/aus, also für boolesche Optionen
- Listenfeld: vertikale Auswahlliste bei der i.d.R. mehrere Werte zu sehen sind und auch ausgewählt werden können
- Rechteck: Grafikelement

- Kontrollkästchen: im Grunde nur ja/nein, also für boolesche Optionen
- ungebundenes Objekt: hat Vor- und Nachteile, für mich überwiegen die Nachteile
- Optionsfeld: sog. *radio buttons* in Kombination mit der Optionsgruppe
- Unterformular: erlaubt die Darstellung von transitiv abhängigen Daten in einem eigenen Formular, z. B. die Funde zu einem Befund
- gebundenes Objektfeld: z. B. für OLE-Objekte, hat Vor- und Nachteile, für mich überwiegen die Nachteile
- Bild: verweist auf eine Grafik, z. B. Logo, auch bei einer eingebetteten Grafik (Teil der DB) bleibt der gesamte Pfad zum geladenen Bild unter der Eigenschaft „Bild“ unnötigerweise erhalten. Wenn Sie Bilder in einer DB verwalten möchten, können Sie die Pfadangabe zum Bild als Text aus einem Textfeld dem leeren Bildrahmen zuweisen, dies ist deutlich effizienter als OLE

4.2.2 Eigenschaften des Formulars

Ein Formular hat viele Eigenschaften. Einige wichtige seien kurz genannt, wechseln Sie hierzu innerhalb des Eigenschaftenblattes auf das Register „Alle“.

- Datensatzquelle: Eine Tabelle oder eine Abfrage, mit [...] können Sie Daten gezielt in einer Abfrage zusammenstellen
- Beschriftung: Titel des Formulars im oberen Fensterrahmen und auf dem Reiter in der Taskleiste von Windows. Damit ist diese Beschriftung sinnvoller als ein opulenter Formularkopf.
- Standardansicht: Einzelnes Formular (der gesamte Detailbereich stellt nur einen DS zur Zeit dar), Endlosformular (der Detailbereich wird über das gesamte Formular je DS wiederholt), Datenblatt (Tabellenansicht ohne Layout)
- Größe anpassen: „ja“ (Standard), passt die Größe des Formulars an den verwendeten Bereich an.
- Daten eingeben: (ziemlich weit unten oder Register „Daten“), heißt nur (!) Daten eingeben, also in der Regel „nein“.

4.2.3 Ein Formular bearbeiten

Der Text geht von dem automatisch generierten Formular (Abb. 4) aus, öffnen Sie dieses in der Entwurfsansicht und öffnen Sie das Eigenschaftenblatt.

Wechseln Sie im Eigenschaftenblatt zum Formular und ändern Sie die Beschriftung in „Befund- u. Funderfassung“.

Im Formularkopf ändern Sie den Text des Bezeichnungsfeldes in „Befund- und Funderfassung“, den Schriftgrad auf 14, ein Doppelklick auf den Rahmen passt die Größe an, positionieren Sie das Feld links oben, passen Sie die Größe des Formularkopfes an.

bef_id_Bezeichnungsfeld: löschen.

Feld bef_id: Schriftgrad: 9, Breite: 2 cm, Doppelklick auf den Rahmen, in die rechte obere Ecke schieben, auf dem Register Daten Aktiviert: nein

Löschen Sie n_sort_Bezeichnung und deaktivieren Sie n_sort analog zu bef_id.

Tragen Sie eine ordentliche Beschriftung in allen Bezeichnungsfeldern ein. Markieren Sie alle Bezeichnungsfelder und passen Sie die Größe mit einem Doppelklick an. Formularentwurfstools → Anordnen → Ausrichtung → rechtsbündig. Markieren Sie alle Datenfelder, Eigenschaftenblatt, Register Format, Links: 3,5 cm. Wenn Sie ein Datenfeld und Bezeichnung werden stets gleichzeitig verschoben, um ein Feld einzeln zu verschieben fassen Sie es am grauen Viereck links oben. Schieben Sie alle Felder übersichtlich zurecht (Abb. 6).

Abb. 6 Formular zur Befund- und Funderfassung in der Entwurfsansicht.

Passen Sie für folgende Felder die Eigenschaften an: bef_id (In Reihenfolge nein), bef_nr (Bildlaufleiste keine, Eingabetastenverhalten Standard), nr_sort (In Reihenfolge nein), flaeche (Bildlaufleiste keine, Eingabetastenverhalten Standard), bef_typ (Textausrichtung linksbündig), Datierung (Textausrichtung linksbündig), beschreib (Eingabetastenverhalten Standard), dm (Bildlaufleiste keine, Eingabetastenverhalten Standard), tiefe (Bildlaufleiste keine, Eingabetastenverhalten Standard). Kontrollieren Sie die Aktivierreihenfolge. Kopieren Sie für jedes Datenfeld im Register Andere den Infotext aus Statusleistertext in Steuerelementtip-Text. Speichern Sie den aktuellen Stand.

4.2.4 Ein bisschen VBA

Markieren Sie das Feld bef_nr und wählen Sie im Eigenschaftenblatt, Register „Ereignis“, „Nach Aktualisierung“ rechts [...], anschließend „Code-Generator“ [OK]. Es öffnet sich das VBA-Fenster und in diesem ein weiteres für die Prozeduren des aktuellen Formulars (Abb. 7).

Kopieren Sie folgenden Code in das Fenster Form_f_ed_befunde für das aktuelle Formular zwischen die Zeilen Private Sub bef_nr_AfterUpdate () und End Sub.

```
'die folgende Prozedur wird nach jeder Aktualisierung des Feldes bef_nr ausgeführt.
On Error GoTo err_bef_nr_AfterUpdate 'Wenn ein Fehler auftritt gehe zu der
    Sprungmarke
```

```
Dim strBefNr As String 'definiert eine Variable vom Typ String
Dim intSortNr As Long 'definiert eine Variable vom Typ Long Integer
Dim intBuchstabe As Integer
```

```
strBefNr = Me!bef_nr 'weist der Variable den Feldwert des aktuellen Formulars zu
'Sie erinnern sich an die Aktualisierungsabfrage in Kap. 3.2?
```

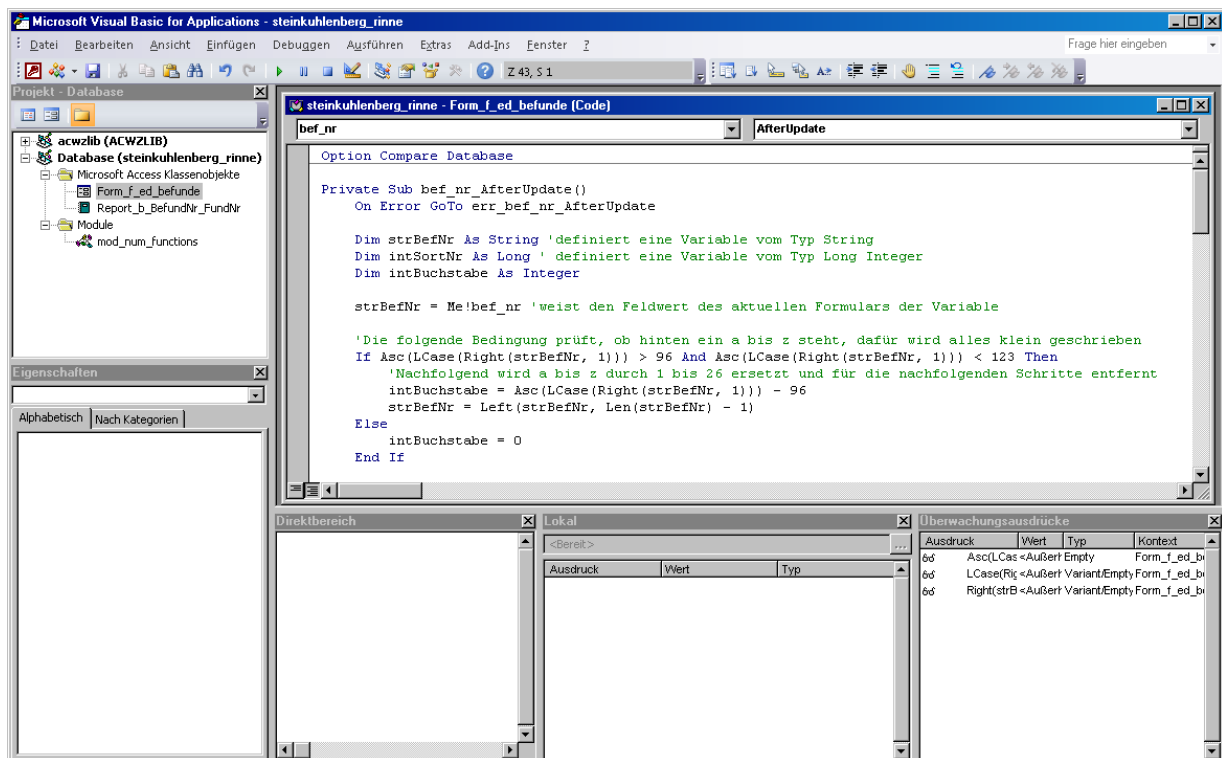


Abb. 7 VBA-Fenster mit dem Fenster der Prozeduren für das aktuelle Formular f_ed_befunde.

```

'Die folgende Bedingung prüft, ob hinten ein a bis z steht, dafür wird alles klein
geschrieben
If Asc(LCase(Right(strBefNr, 1))) > 96 And Asc(LCase(Right(strBefNr, 1))) < 123
Then
    'Nachfolgend wird a bis z durch 1 bis 26 ersetzt und für die nachfolgenden
    Schritte entfernt
    intBuchstabe = Asc(LCase(Right(strBefNr, 1))) - 96 'Buchstabe in Zahl
    strBefNr = Left(strBefNr, Len(strBefNr) - 1) 'Zeichenkette ohne Buchstabe
Else
    intBuchstabe = 0
End If

intSortNr = CInt(strBefNr) * 100 + intBuchstabe 'Zahl um zwei Zeichen nach links
rücken und Buchstabe „addieren“

Me!nr_sort = intSortNr 'Dem Formularfeld die Sortierungsnummer zuweisen

exit_bef_nr_AfterUpdate: 'Sprungmarke
Exit Sub 'Verlasse die Prozedur

err_bef_nr_AfterUpdate: 'Sprungmarke für Fehler
MsgBox Err.Number & vbCrLf & Err.Description 'Gibt die Fehlernummer- und
Beschreibung aus
Resume exit_bef_nr_AfterUpdate 'mach weiter mit der folgenden Sprungmarke

```

Ergänzenden Anmerkungen:

Die Funktion Asc(Zeichenkette) gibt den ASCII-Zeichencode zurück, z.B. Asc(a) → 65.

Die Funktion Right(Zeichenkette, n) gibt n Zeichen vom rechten Rand der Zeichenkette zurück.

Die Funktion Left(Zeichenkette, n) gibt n Zeichen vom linken Rand einer Zeichenkette zurück.

Die Funktion Mid(Zeichenkette, n, i) gibt ...? ... ab dem n-ten Zeichen i Zeichen zurück.

Die Funktion CInt (Zeichenkette) wandelt eine Zeichenfolge aus Ziffern in eine Ganzzahl um.

Die Funktion Len(Zeichenkette) gibt die Anzahl der Zeichen einer Zeichenkette zurück.

Die Funktion LCase(Zeichenkette) gibt eine Zeichenkette in Kleinbuchstaben zurück.

Wenn der Cursor in einer Funktion steht und sie <F1> drücken erhalten Sie eine ausführliche Hilfe.

Therme werden von innen nach außen ausgewertet, also die innerste Klammer zuerst. Das machen wir mal mit der ersten Bedingung des ersten Elself mit der Vorgabe strBefNr = ‚15b‘:

```
Asc(LCase(Right(strBefNr, 1))) > 64
```

1. Nimm von ‚15b‘ das erste Zeichen vom rechten Rand → b
2. Wandle b in einen Kleinbuchstaben → b
3. Gib von ‚b‘ den zugehörige ASCII-Zahlencode zurück → 98.
4. Ist 98 > (größer als) 96 → Wahr

Die zugehörigen Aktionen nach „Then“ lautet:

```
intBuchstabe = Asc(LCase(Right(strBefNr, 1))) – 96
```

```
strBefNr = Left(strBefNr, Len(strBefNr) - 1)
```

Die erste Anweisung subtrahiert von der soeben ermittelten Codezahl 96, unser ‚b‘ ergibt also 2. Die zweite Anweisung substituiert die Befundnummer mit Suffix durch die Befundnummer ohne Suffix, indem vom linken Rand die Länge der Befundnummer minus 1 gefordert wird.

Abschließend wird für alle Fälle noch folgende Anweisung durchgeführt:

```
intSortNr = CInt(strBefNr) * 100 + intBuchstabe
```

Der rechte Teil der Gleichung (eigentlich eine Zuweisung) besteht 1. aus einer Multiplikation und 2. einer Addition (Punktrechnung vor Strichrechnung)

1. Wandle ‚15‘ in eine Zahl um → 15
2. multipliziere 15 mit 100 → 1500
3. addiere den Buchstabenwert (2) → 1502

Betrachten Sie bitte den vorangehenden Code, lesen Sie die mit Hochkomma ausgezeichneten, grün dargestellten Kommentare und geben Sie sich bitte 5 Minuten Zeit. Entscheiden Sie bitte erst dann, ob Sie dieses VBA-Fenster zum letzten mal gesehen haben.

Ziel war es, aus den Befundnummern mit dem Buchstabensuffix automatisch eine Zahl für die Sortierung zu erstellen. Da die Programmierung sehr simpel ist und nach ‚z‘ eventuell ‚aa‘ als Suffix kommen könnte oder irgend jemand garantiert auf die Idee kommt, die Grube Befund 100 als ‚G100‘ zu bezeichnen (s.u.), ist der Lösungsansatz nicht wirklich gut umgesetzt. Mögliche Alternativen sind das Auffüllen des Textfeldes mit „0“ zu „000015a“ oder die Trennung in zwei Felder, also Zahl und Text. Die Befundnummer „15a“ in der Grabungsdokumentation ist aber schon etwas anderes als „000015a“ und wer sagt Ihnen, dass es nicht noch den Befunde „111115a“ gibt? Wenn ein neuer Befund vorliegt, kriegt er eine neue Nummer; dieses „er gehört da ein bisschen zu aber nicht so richtig“ ist lau. Also am besten *a priori* vermeiden und sich dezidiert entscheiden!

4.2.5 Ein Unterformular erstellen

Nutzen Sie den Assistenten, wählen Sie t_funde und das einspaltige Layout. Markiere Sie alle Felder ändern Sie die Schriftgröße auf 10 pt (Formularentwurfstools → Format → Schriftart). Reduzieren Sie die Breite des Formulars auf 16,4 cm durch ziehen des rechten Randes mit der Maus und wechseln Sie die Eigenschaft des Formulars auf Endlosformular. Löschen Sie fund_id und bef_nr vollständig, und ordnen Sie die übrigen Felder in horizontalen Reihen in der Folge, wie diese im Katalog von Hille (1986) auftreten. Ihr Formular könnte nun etwa so aussehen (Abb. 8).



Abb. 8 Entwurfsansicht des Unterformulars Funde.

Wechseln Sie zum Entwurf des vorigen Formulars f_ed_befunde. Wählen Sie das Steuerelement Unterformular aus und ziehen Sie einen Rahmen im gesamten unteren freien Bereich des Formulars. Im folgenden Fenster wählen Sie „Vorhandenes Formular ...“ und das soeben erstellte Formular f_ed_befunde_funde aus. Das folgende Fenster verstehe ich besser, wenn ich oben „Eigene definieren“ auswähle, dann muss ich unten die Schlüsselfelder für die Relation auswählen. In diesem Fall also auf beiden Seiten „bef_nr“. Das Steuerelement Unterformular muss noch einen Namen erhalten, ich kürze hier gerne etwas ab „uf_funde“ (Abb. 9).

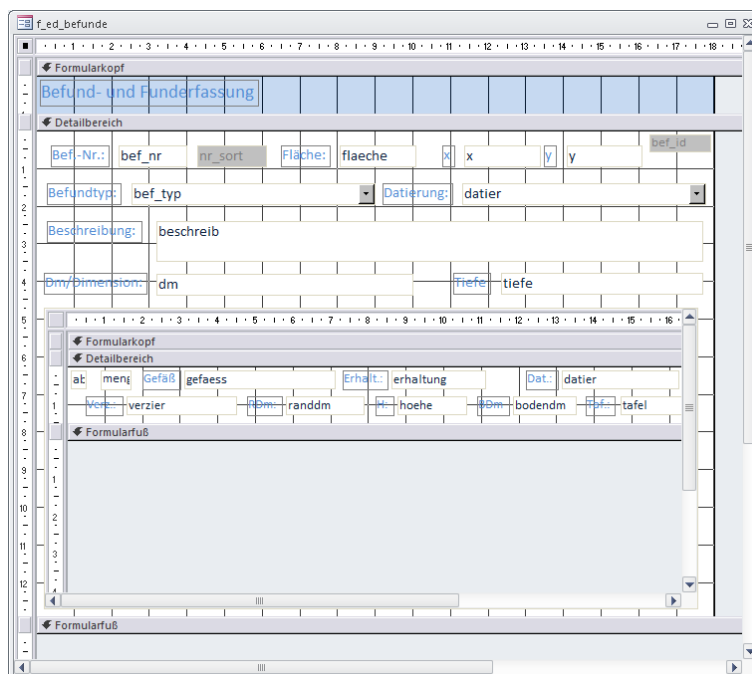


Abb. 9 Formular zur Befund- und der untergeordneten Funderfassung.

Die Informationen beider Formulare sind nun verknüpft über das Feld bef_nr. Dieses Feld ist im Unterformular somit überflüssig für den Nutzer, denn es muss dem Hauptformular entsprechen. In der Liste von Hille (1986) steht ja auch nicht jedes mal die Befundnummer bei den Funden. Die Befundnummer wird durch die gesetzte Relation automatisch in der Tabelle t_funde eingetragen, also ein Problem weniger.

5 Berichte

Berichte dienen dem Druck von Daten. Neben schlichten Listen lassen sich mit etwas Geschick auch zweiseitige, weitgehend fertige Kataloge oder umfassende Datenblätter erstellen. Ein Export des Ergebnisses zu MS Word *via* rtf-Datei ist möglich, das Ergebnis ist aber nicht wirklich überzeugend und stellt *de facto* keine Alternative dar. Im Gegensatz zu MS Access 2003 bietet 2010 auch den

Export in eine txt-Datei, dies ist für zahlreiche Datenbankanwendungen sehr erfreulich (z. B. Code mit ¹⁴C-Daten für Kalibrationsprogramme, KML-Dateien für Fundplätze, reiner Katalogtext für das Layout in einem anderen Programm etc.).

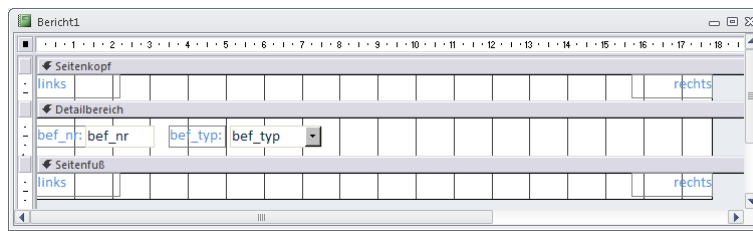


Abb. 10 Seitenlayout in MS Access 2010.

Eine kurze Vorbemerkung zur Gestaltung des Berichtes in Bezug zum Papierformat und dem Einrichten der Seite. Im Berichtsentwurf sehen Sie in der Regel Seitenkopf, Detailbereich und Seitenfuß in einer Breite, die bei DIN A4 zwischen 16 und 19 cm betragen sollte. Es handelt sich um die Breite des Berichtes, die Abstände zum Papierrand werden in der Seiteneinrichtung definiert. Die Berichtsbreite wird auch in der Eigenschaft „Breite“ auf dem Eigenschaftensblatt des Berichtes angegeben. Analog ist die Höhe des Detailbereiches die Höhe für jeden einzelnen Datensatz, nicht die Höhe für das Papier. Etwas verwirrend ist die Arbeit mit Spalten, die Spaltenbreite entspricht nicht der Berichtsbreite, sondern sollte um mindestens den Faktor 0,5 kleiner sein.

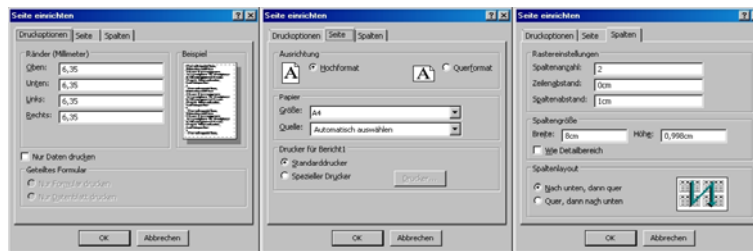


Abb. 11 Seite einrichten für einen Bericht.

Das Einrichten der Seite erfolgt je Bericht mit Berichtentwurfstools → Seite einrichten → Seite einrichten. Das Fenster hat drei Register für „Druckoptionen“, „Seite“ und „Spalten“ (Abb. 11). Bei Access 2003 und älter finden Sie diese Einstellungen analog zu MS Word unter Datei → Seite einrichten. Die ersten Register sind selbsterklärend, das dritte Register definiert die Spalten und Zeilen (letzteres für Etiketten). Hier finden Sie die Spaltengröße, die wie zuvor erwähnt in der Summe von Spalten und Spaltenabstand etwas kleiner als die Berichtsbreite sein sollte. Mit Berichtentwurfstools → Entwurf → Ansicht → Seitenansicht erhalten Sie eine Seitenvorschau (Abb. 12). Die Seitenvorschau verlassen Sie mit <Esc> oder dem rot hinterlegten „X“ rechts oben im Menü.

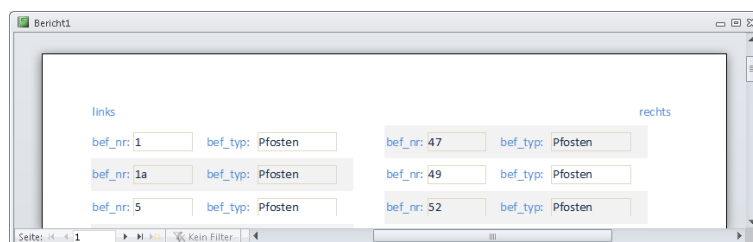


Abb. 12 Seitenvorschau zu einem zweispaltigen Bericht.

5.1 Ein Bericht mit dem Berichts-Assistent

Mit Erstellen → Berichts-Assistent erstellen Sie schnell eine Auflistung von Daten. Die Vorgehensweise entspricht weitgehend dem Formularassistent, auch die spätere Gestaltung weist zahlreiche Analogien auf. Nach der Auswahl der Datenquelle, hier t_befunde, können Sie Gruppierungsebenen bestimmen, hier z.B. flaeche. Im folgenden Fenster wählen Sie bef_nr als erstes Sortierkriterium, das folgende Layout in diesem Fall bitte belassen und abschließend im letzten Fenster einen sinnvollen Namen vergeben, z.B. b_befunde. Auch in diesem Fall überzeugt das Ergebnis nicht wirklich (Abb. 13).

Für die Gestaltung eines Layouts sei noch auf folgende Optionen hingewiesen.

- Der Assistent bietet noch Zusammenfassungsoptionen für die Gruppierung, diese können als Vorlage für eigene Anweisungen genutzt werden.
- Sehen Sie sich die automatisch generierten Felder im Seitenfuß im Entwurfsmodus an.
- Sie können Unterberichte erstellen und diese Einbinden, analog den Formularen. Dies ermöglicht ein Komplexes Layout, z. B. spezifische Fundlisten für Materialgruppen je Befund.
- Im Eigenschaftenblatt des Berichtes, „Datensatzquelle“ kann auch eine umfangreiche Abfrage über mehrere Tabellen erstellt werden.
- Abfragen erlauben die Verkettung der Informationen mehrere Felder in einem Textfeld, hierdurch wird das Layout für Felder mit variabler Länge deutlich erleichtert. Ein Beispiel wäre:

- Koord: [x] & "/" & [y]
- Beschr: Wenn([beschreib];[beschreib] & ". ";"") & Wenn([dm];"Dm.: " & [dm] & " cm. ";"") & Wenn([tiefe];"T: " & [tiefe] & " cm. ";"")

Der erste Fall stellt beide Koordinaten in einem Textfeld durch Schrägstrich getrennt dar. Das zweite Beispiel prüft jeweils ob ein Wert vorliegt und stellt diesem dann eine Erläuterung voran und die Maßeinheit mit Satzzeichen nach. Statt drei Felder mit jeweils variabler Länge erhalten Sie nur ein Textfeld und sparen sich auch noch die Feldbezeichnung.

Im Vergleich zur oftmals publizierten Tabellen aus Excel oder Calc können Sie mit Access einen vollständig gestalteten Katalog generieren.

- 📘 Nicht alles was möglich ist, ist auch gut. Das automatische Layout der Berichte nutzt Farben, das sieht am Monitor gut aus, ist nach dem Ausdruck in Schwarz-Weiß aber schlechter zu lesen. Arbeiten Sie im Layout mit Fett, sparsam mit der Schriftgröße und verwenden Sie nur eine Schrifttype.

flaeche	bef_nr	f_id_sort	x	y	bef_typ	datier	beschreib	dm	tiefe
158	95	###	####	####	Pfofen			40	20
126	73	###	####	####	Pfofen			20	8
130	74	###	####	####	Pfofen			28	32
132	75	###	####	####	Pfofen			20	9
133	76	###	####	####	Pfofen			18	10

Abb. 13 Ein fertiger Bericht des Assistenten als Liste.

5.2 VBA im Bericht

Für manche Probleme muss man doch in die Programmierung. Sie haben eine eigene Tabelle mit den Fundnummer, Fotonummer oder sonst einer tabellarischen Aufstellung. Nun brauchen Sie keine vertikale Tabelle, sondern möchten eine mit Kommata getrennte Liste als Fließtext generieren. Wir spielen das im Folgenden mal durch, vorab aber kurz der traditionelle Fußweg: Aus der Gesamtliste 1. exzerpiere alle Befundnummern, 2. suche alle Fundnummern des ersten Befundes, 3. sortiere diese nach Maßgabe und 4. schreibe diese mit Kommata getrennt in ein Textfeld. Wiederhole Schritte 2 bis 4 für alle weiteren Befunde der Liste aus Schritt 1. Das ist fast 1:1 übertragbar.

- 🔵 VBA wird seit 2007 nicht weiter entwickelt, es wird also irgendwann aussterben. Es bleibt aber eine leicht zu erlernende Programmiersprache für Office-Anwendungen.

Erstellen Sie einen Bericht in der Entwurfsansicht: Breite 17 cm, Seitenränder mit 20, 20,20, 15. Im Seitenkopf könne Sie ein Textfeld platzieren mit „Fundnummern je Befund“. Rufen Sie die Eigenschaften des Berichtes auf, wechseln Sie zum Register „Ereignisse“ und erstellen Sie mit [...] eine VBA-Prozedur für das Ereignis „Beim Öffnen“.

```
Private Sub Report_Open(Cancel As Integer)
'erstellt eine neue Tabelle einer Spalte für die Befundnummer und eine Spalte für
die Liste der Fundnummern
On Error GoTo err_Report_Load
    Dim str_tmptable 'Name für eine Tabelle mit der zu erstellenden
Fundnummernliste
    Dim rs_befunde As DAO.Recordset 'Recordset mit tabellarischer Liste der Befunde
    Dim rs_funde As DAO.Recordset 'Recordset mit der tabellarische Liste der Funde
    Dim str_BefNr, str_Fundliste As String 'zwei Variablen für die anzufügenden
Daten
    Dim str_SQL As String 'Die Fundliste jedes Befundes wird mit SQL in die neue
Tabelle geschrieben.

    If DCount("*", "MSysObjects", "Name='t_tmp_fundnr'") > 0 Then
        'wenn die Tabelle schon existiert, dann löschen
        DoCmd.DeleteObject acTable, "t_tmp_fundnr"
    End If

    'Neue und leere Tabelle erstellen
    DoCmd.RunSQL "CREATE TABLE t_tmp_fundnr (befnr TEXT(255), fundnr MEMO);"

    'Dem Recordset wird eine SQL-Abfrage zugewiesen, die eine gruppierte Liste der
Befundnummern erstellt.
    Set rs_befunde = CurrentDb.OpenRecordset("SELECT t_funde.bef_nr FROM t_funde
GROUP BY t_funde.bef_nr ORDER BY t_funde.bef_nr;")
    'Wenn der Recordset Daten enthält
    If rs_befunde.RecordCount > 0 Then
        'tu bis zum Ende des Recordset (End Of File)
        Do Until rs_befunde.EOF
            'Der Variable die Befundnummer des Recordset zuweisen
            str_BefNr = rs_befunde!bef_nr
            'Dem Recordset wird eine SQL-Abfrage zugewiesen, die eine sortierte
Tabelle der Fundnummern dieses Befundes erstellt.
            Set rs_funde = CurrentDb.OpenRecordset("SELECT t_funde.fund_id FROM
t_funde WHERE t_funde.bef_nr = '" & str_BefNr & "' ORDER BY t_funde.fund_id;")
            'Wenn der Recordset Daten enthält
            str_Fundliste = "" 'Fundnummernliste leeren
            If rs_funde.RecordCount > 0 Then
                Do Until rs_funde.EOF
                    'Wenn die Liste leer ist fang an, ansonsten erst ein Komma und
dann die Fundnummer
                    If str_Fundliste = "" Then
                        str_Fundliste = rs_funde!fund_id
                    Else
                        str_Fundliste = str_Fundliste & ", " & rs_funde!fund_id
                    End If
                Loop
            End If
        Loop
    End If
```

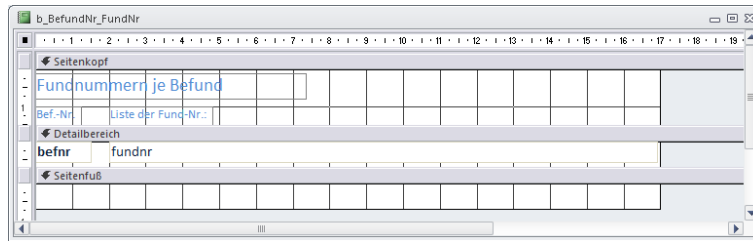



Abb. 14 Der Bericht für die Fundnummernliste im Layout

```

        'und nicht vergessen, einen Schritt weiter!
        rs_funde.MoveNext
    Loop 'gehe zurück
End If
'Recordset ordentlich schließen und löschen
rs_funde.Close
Set rs_funde = Nothing
'Nun ist die Fundnummernliste für diesen Befund fertig und die
Anfügeabfrage wird erstellt.
str_SQL = "INSERT INTO t_tmp_fundnr (befnr, fundnr) SELECT '" &
str_BefNr & "', '" & str_Fundliste & "';"
DoCmd.SetWarnings False 'Access will sonst wissen, ob es einen
Datensatz anfügen soll.
DoCmd.RunSQL (str_SQL) 'SQL-Anfügeabfrage ausführen
DoCmd.SetWarnings True 'und wieder einschalten
rs_befunde.MoveNext 'und zum nächsten Befund!
Loop 'für Do Until rs_befunde.EOF
End If 'end if für rs_befunde.recordcount > 0
rs_befunde.Close 'Recordset schließen
Set rs_befunde = Nothing 'Recordset löschen

'Nun ist die Datengrundlage fertig und wird dem Bericht zugewiesen
Me.RecordSource = "t_tmp_fundnr"

exit_Report_Load:
Exit Sub

err_Report_Load:
MsgBox Err.Number & vbCrLf & Err.Description
Resume exit_Report_Load
End Sub

```

Wählen Sie die temporär erstellte Tabelle t_tmp_fundnr als Quelle des Berichtes und positionieren Sie die beiden Felder befnr und fundnr in Ihrem Bericht. Sie finden die Tabelle nicht und sind ohne die üblichen Auswahlfelder etwas unsicher? Kopieren Sie den SQL-Code zum erstellen der Tabelle t_tmp_fundnr aus dem VBA-Code in eine leere Abfrage in der SQL-Ansicht und führe Sie diese einfach aus. Nun können Sie auf die Daten in der gewohnten Weise zugreifen. Der fertige Berichtsentwurf könnte aussehen wie in Abb. 14. Auf das Ereignis „Beim Deaktivieren“ sollten Sie die temporäre Tabelle auch wieder löschen, der Code dafür steht bereits im vorangehenden Beispiel.

```

Private Sub Report_Deactivate()
'lösst erst die Datenquelle aus dem Bericht um diese dann anschließend zu löschen
Me.RecordSource = ""
If DCount("*", "MSysObjects", "Name='t_tmp_fundnr'") > 0 Then
'wenn die tabelle schon existiert, dann löschen
DoCmd.DeleteObject acTable, "t_tmp_fundnr"
End If
End Sub

```

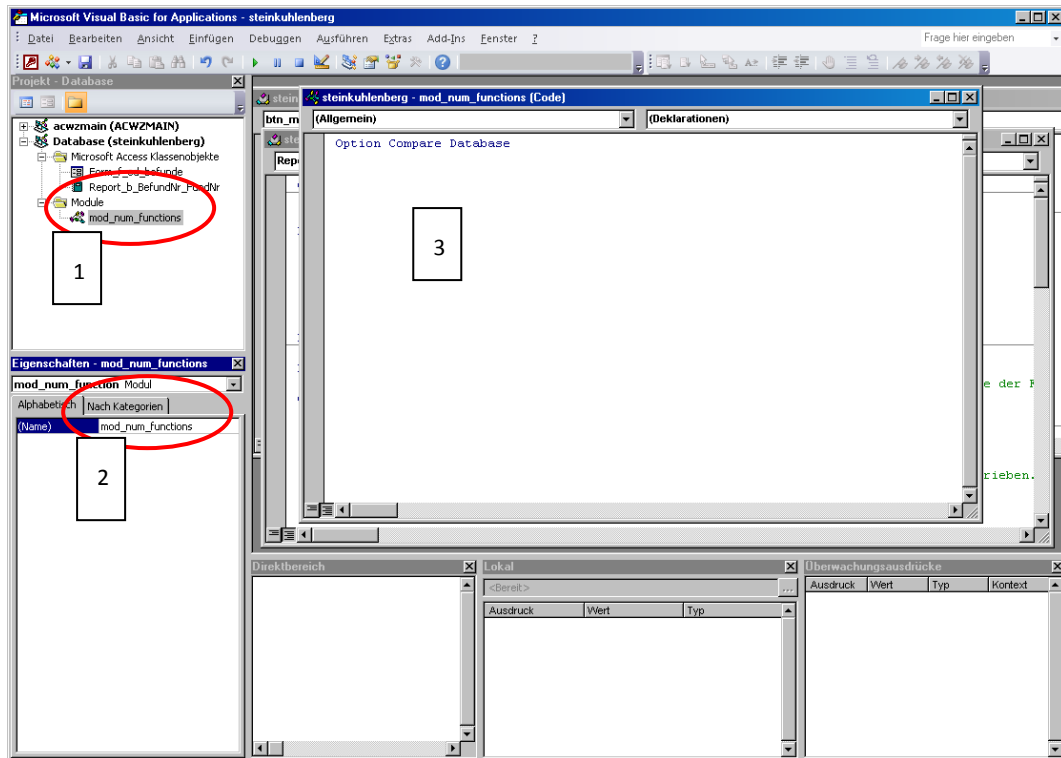


Abb. 15 VBA-Fenster mit dem Fenster für ein neues Modul.

6 Module

Vor allem im letzten Kapitel haben Sie gemerkt, eine Datenbank ist viel mehr als eine Sammlung von Tabellen und eventuell ein jeweils zugehöriges Formular. Viele Prozesse können automatisiert werden um eine Normierung zu erreichen oder um sich wiederholende Arbeiten effizienter zu gestalten.

- 📘 Entscheiden Sie aber bitte, ob aktuell der Weg das Ziel ist oder doch lieber das Ziel selbst.
- 📘 Ziel dieses Kapitels ist ein kursorischer Einblick mit einem schnellen Erfolg, um Ihnen die Möglichkeiten aufzuzeigen.

Module beinhalten Funktionen, sagen wir mal kleine Programme, die regelmäßig an unterschiedlichen Orten gebrauchte Aufgaben erledigen. Einige gängige Funktionen haben Sie bereits kennengelernt (Kap. 4.2.4).

Die Gruben im Katalog vom Steinkuhlenberg (Hille 1986) wird mit römische geschriebenen Zahlen benannt. Das ist nun leider einmal so. Wegen des ergänzenden „a“ bei der arabischen Zählung der Pfosten ist der Befundname bereits als Textfeld definiert, der stringenten Übernahme der Vorgabe steht formal also nichts im Weg. Die Sortierung im Hintergrund sollte aber doch irgendwie „schön“ sein, deshalb wird nachfolgend jede römische Zahl automatisch als arabische Zahl in das Feld nr_sort geschrieben. Wir müssen also eine kleines „Programm“ schreiben, welches diesen Vorgang für uns automatisch erledigt. Ist gar nicht so schwer, jeder gültige Buchstabe wird durch seinen Wert ersetzt, sind die letzten beiden Wert kleiner als der aktuelle, werden sie subtrahiert, sonst addiert.

Wir erstellen nun also ein eigenes Modul und fügen diesem eine erste Funktion zur Konvertierung römischer Zahlen hinzu. Im Formular der Befunderfassung rufen wird dann nur noch die vorab geschriebene Funktion auf, übergeben dieser unsere römische Zahl und erhalten die arabische als

Rückgabe. Starten Sie hierzu die VBA-Anwendung von Access, ein Weg führt über das Menü: Erstellen → Modul, ein anderer startet direkt diese Anwendung mit <Alt> + <F11> und fügt dann mit Einfügen → Modul ein neues Modul hinzu. Das neue Modul wird automatisch „Modul1“ benannt und sollte sinnvoll umbenannt werden, z. B. „mod_num_functions“. Markieren Sie hierfür das Modul links im Projekt-Fenster (1 in Abb. 15) und ändern Sie den Namen im Eigenschaftfenster unterhalb (2 in Abb. 15). Das zugehörige Modulfenster sollte rechts aktiv und zu sehen sein (3 in Abb. 15). Kopieren Sie den folgenden Code in das Modul unter den Eintrag Option Compare Database und speichern Sie.

```
'Diese Funktion wandelt römische in arabische Zahlen um
Public Function Rome2Arab(ByVal strRome As String) As Integer
    'An die Funktion wird ein String übergeben, der Rückgabewert ist vom Typ
    Integer
    Dim i, intRome, intRome2, intRomeSum As Integer
    ' i brauchen wir zum Zählen der Ziffern, den Rest für Werte der Buchstaben

    strRome = UCase(strRome) 'Upper Case, falls jemand klein geschrieben hat
    intRomeSum = 0
    intRome = 0
    intRome2 = 0

    For i = 1 To Len(strRome) 'i zählt jeweils ein Zeichen in strRome weiter
        Select Case Mid(strRome, i, 1) 'nimm das Zeichen an Position i und wähle
            Case Is = "M"
                intRome = 1000
            Case Is = "D"
                intRome = 500
            Case Is = "C"
                intRome = 100
            Case Is = "L"
                intRome = 50
            Case Is = "X"
                intRome = 10
            Case Is = "V"
                intRome = 5
            Case Is = "I"
                intRome = 1
            Case Else
                'und immer an die ultimative Alternative denken
                Rome2Arab = 0
                Exit Function
        End Select
        'Einzelwerte addieren
        If intRome > intRome2 Then
            intRomeSum = intRomeSum - intRome2 * 2 + intRome
        Else
            intRomeSum = intRomeSum + intRome
        End If
        intRome2 = intRome
    Next i
    Rome2Arab = intRomeSum
End Function
```

Wechseln Sie im VBA-Fenster zum Code des Formulars f_ed_befunde, klicken Sie hierfür auf den entsprechenden Eintrag Form_f_ed_befunde im Projekt-Fenster. Da steht ja nun für das Ereignis bef_nr_AfterUpdate() ja schon die Sache mit dem Suffix drin. Was ist zu tun? Genau diese Entscheidung müssen wir nun eindeutig anweisen und zwar so, dass es stets eine Antwort gibt. Ich habe inzwischen gesehen, dass im Katalog Pfosten wie die Flächen arabisch gezählt werden und zur Unterscheidung ein „P“ beziehungsweise ein „F“ vorangestellt wird. Etwas später gibt es noch Grabenschnitte, die innerhalb des Jahres mit A, B, C,... bezeichnet werden. Ich entscheide mich wie folgt: Die vorhandene erste Anweisung bleibt, also: 1. Wenn am rechten Rand ein Buchstabe steht, nimm dessen Position im Alphabet, und kürze die Befundnummer. Danach wird die neue

Befundnummer geprüft: 2. Ist das erste Zeichen eine Zahl oder ist es ein „P“ oder ist es ein „F“ oder ist es ein „GS“? Je nach dem addiere zur Befundnummer ohne diese Buchstaben einen festen Wert. Im Ergebnis sind Pfofen 10.000er, Flächen 20.000er und Grabenschnitte 30.000er. Wenn keine der Bedingungen zutrifft übergib den Wert an die Funktion Rome2Arab(). Die Funktion Rome2Arab() gibt eine „0“ zurück, wenn ein ungültiger Buchstabe übergeben wird, damit wird alles, was nicht verarbeitet werden kann mit 0 zurückgegeben.

Fügen Sie nach ersten IF-THEN-ELSE Anweisung bitte folgende Bedingung ein:

```
'Die folgende Bedingung prüft, ob links eine Zahl, ein P für Pfofen, ein F für
  Fläche, ein GS für Grabenschnitt, ...
If Asc(Left(strBefNr, 1)) > 47 And Asc(Left(strBefNr, 1)) < 58 Then
  MsgBox "Gruben bitte römisch, Pfofen mit 'P', Fläche mit 'F' und
    Grabenschnitt mit GSJahr eingeben.", vbOKOnly + vbInformation
  intSortNr = CInt(strBefNr)
ElseIf Left(strBefNr, 1) = "P" Or Left(strBefNr, 1) = "p" Then
  'Die Pfofen werden nun als 10.000er gezählt
  intSortNr = CInt(Mid(strBefNr, 2, Len(strBefNr) - 1)) + 10000
ElseIf Left(strBefNr, 1) = "F" Or Left(strBefNr, 1) = "f" Then
  'Die Flächen werden nun als 20.000er gezählt
  intSortNr = CInt(Mid(strBefNr, 2, Len(strBefNr) - 1)) + 20000
ElseIf Left(strBefNr, 2) = "GS" Or Left(strBefNr, 2) = "gs" Then
  'Die Grabenschnitt werden nun als 30.000er gezählt
  intSortNr = CInt(Mid(strBefNr, 2, Len(strBefNr) - 2)) + 30000
Else
  'Die Gruben mit de römischen Zahlen werden einfach gezählt
  intSortNr = Rome2Arab(strBefNr)
End If

'Nun noch die Suffixbuchstaben anhängen und dafür alles um 2 Stellen nach links
  schieben
Me!nr_sort = intSortNr * 100 + intBuchstabe
```

Und löschen Sie nun den Rest bis zur Sprungmarke exit_bef_nr_AfterUpdate:

Sieht doch super aufgeräumt und einfach aus.

Ihre nächste eigene Funktion im Modul mod_num_functions könnte is_numeric (strString) lauten. Dazu brauchen wir zwei Funktionen: 1. CInt(String) gibt einen Fehler zurück, wenn String nicht Ganzzahl oder Kommazahl ist. 2. Replace(String, „.“, „,“) ersetzt in einem String alle Punkte durch ein Komma, könnte jemand ja auch 12.3 statt 12,3 schreiben. Wenn also CInt(Replace(Prüfstring, „.“, „,“)) ohne Fehler durchläuft gibt die Funktion ein TRUE zurück, wenn das Ereignis On Error eintritt wird ein FALSE zurückgegeben. Alternative Wege sind möglich.

7 Abkürzungen

DB	Datenbank
DS	Datensatz
RS	Recordset

8 Literatur

Hille 1986: A. Hille, Die Besiedlung des Steinkohlenberges bei Derenburg, Kreis Wernigerode (nach Ausgrabungsergebnissen des Institutes für Ur- und Frühgeschichte der Martin-Luther-Universität Halle/Wittenberg in den Jahren 1960-64 und 1966) (Halle, unpubl. Diplomarbeit 1986).