Ausgabe 02/2023

ACCESS BASICS

DAS ACCESS-MAGAZIN FÜR ALLE,
DIE VON 0 AUF 100 WOLLEN

In diesem Heft:

André Minhorst Verlag

ACCESS BASICS

DAS ACCESS-MAGAZIN FÜR ALLE, DIE VON 0 AUF 100 WOLLEN

Ausgabe 02/2023

Das neue Access [basics] Kompendium 2022 ist bald da!
In Kürze erscheint das neue **Access [basics] Kompendium 2022** mit allen Ausgaben von 2010 bis 2022.

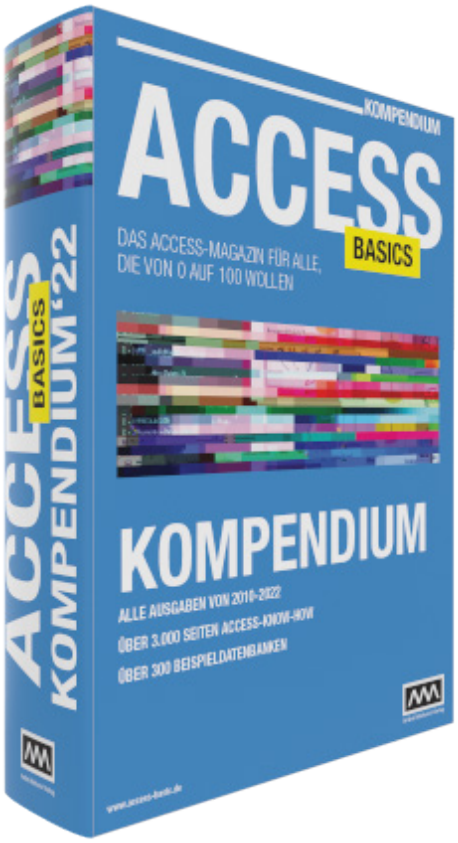
Als aktueller Abonnent erhalten Sie das Kompendium zum Preis von 59 EUR statt 299 EUR. Sie müssen nur bei der Bestellung den Gutscheincode **komp100** eingeben.

Sie erhalten mit dem Kompendium alle Ausgaben in einem praktischen PDF mit über 3.000 Seiten. Außerdem können Sie passend dazu ein Archiv mit allen Beispieldatenbanken der Magazine herunterladen und haben somit alles im Überblick – auch, wenn Sie mal offline sind.

Hier ist der Link zur Bestellung des neuen Kompendiums:

https://shop.minhorst.com/detail/index/sArticle/376/sCategory/76

Viel Spaß beim Lesen!



IMPRESSUM

ACCESS [BASICS] WIRD HERAUSGEGEBEN VON:
André Minhorst Verlag
Borkhofer Straße 17
47137 Duisburg

REDAKTION:
André Minhorst (V.i.S.d.P)
Telefon: 0203/4495577
E-Mail: info@access-basics.de
Internet: www.access-basics.de

Geschäftsführung, Herstellung, Text- und Schlussredaktion, Layout von Magazin und Webseite: André Minhorst
Autor: André Minhorst
Fach- und Sprachlektorat: Carsten Gromberg

ISSN: 2190-8761

Abfragen [basics]: Inkonsistenzen per Assistent suchen

Der Abfrage-Assistent zur Inkonsistenzsuche hat eine etwas andere Aufgabe, als man es sich eventuell vorstellt. Das kommt jedoch darauf an, welche Vorstellung vom Begriff Inkonsistenz im Bereich Datenbanken hat. In diesem Artikel klären wir die Vorstellung des Abfrage-Assistenten zur Inkonsistenzsuche davon und was wir darüber denken und zeigen, wie Du diesen Assistenten dennoch sinnvoll einsetzen kannst – nämlich um schnell Datensätze zu finden, die noch nicht über das Fremdschlüsselfeld einer bestimmten anderen Tabelle verknüpft wurden.

Beispieldatenbank

Die Beispiele dieses Artikels findest Du in der Datenbank **2302_AbfragenBasics_InkonsistenzenPerAssistentSuchen.accdb**.

Was sind Inkonsistenzen?

Nach unserer Idee sind Inkonsistenzen im Datenbankumfeld eher solche Daten in Feldern wie Anrede oder PLZ, die das gleiche bedeuten sollen, aber unterschiedlich geschrieben wurden und somit nicht mehr als eins erkennbar sind. Wenn man also beispielsweise keine Lookuptabelle zum Speichern von Anreden erstellt hat und diese von Hand in das Feld eingibt, kann es passieren, dass man nicht nur Herr und Frau, sondern vielleicht auch mal Herrn einträgt.

Und wenn man dann das Anrede-Feld nutzt, um auf das Geschlecht zu schließen, gelingt das nicht mehr nur durch den Vergleich mit **Herr** oder **Frau**, die Datensätze mit der Anrede **Herrn** fallen dann durchs Raster.

Auch bei PLZs gibt es recht unterschiedliche Schreibweisen – manch einer nutzt noch das offiziell nicht mehr verwendete D-47137 statt nur 47137. Auch hier kann man dann nur noch ein geschränkt nach Adressen in einem bestimmten Postleitzahlenbereich suchen.

Während man das Problem bei der PLZ durch das Prüfen des Formats bei der Eingabe in den Griff bekommen kann, könnte man bei Anreden mit einer Lookuptabelle Inkonsistenzen vermeiden.

Was der Abfrage-Assistent zu Inkonsistenten sagt

Der Abfrage-Assistent meint mit Inkonsistenzen etwas anderes, nämlich Daten in einer Tabelle, die noch nicht von einer anderen Tabelle aus referenziert wurden, also quasi bezüglich dieser Beziehung im »luftleeren Raum« stehende Datensätze. In der Beschreibung gibt der Assistent das Beispiel von Kunden, die noch mit keiner Bestellung verknüpft sind. Es gibt also beispielsweise keinen Datensatz in

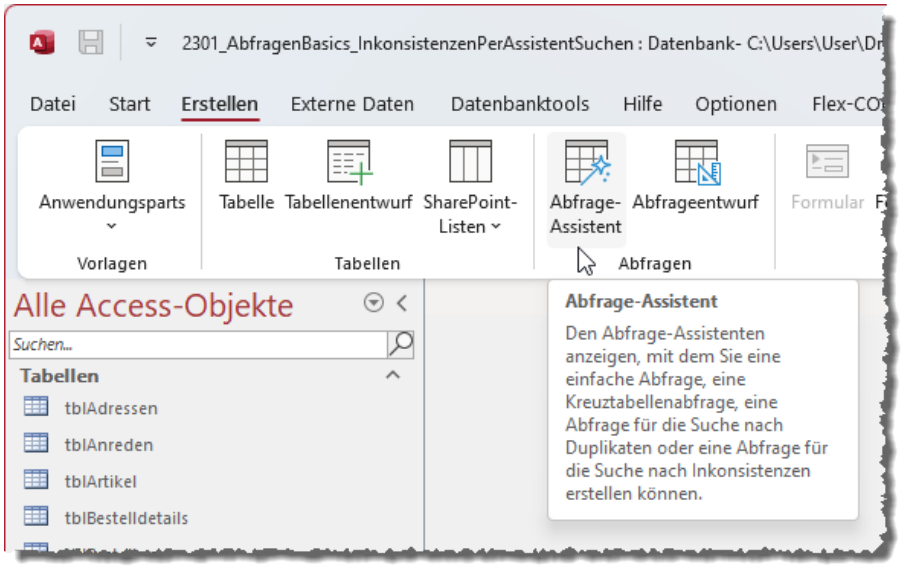


Bild 1: Aufrufen des Dialogs **Neue Abfrage**

der Tabelle **tblBestellungen**, der mit einem bestimmten Kunden verknüpft ist, und solche Kundendatensätze findet der Abfrage-Assistent zur Inkonsistenzsuche für uns.

Abfrage-Assistent zur Inkonsistenzsuche aufrufen

Diesen Assistent rufen wir aus dem gleichen Bereich des Ribbons heraus auf, der auch die Befehle zum Anlegen einer neuen Abfrage anbietet, also unter **Erstellen|Abfragen** unter dem Namen **Abfrage-Assistent** (siehe Bild 1).

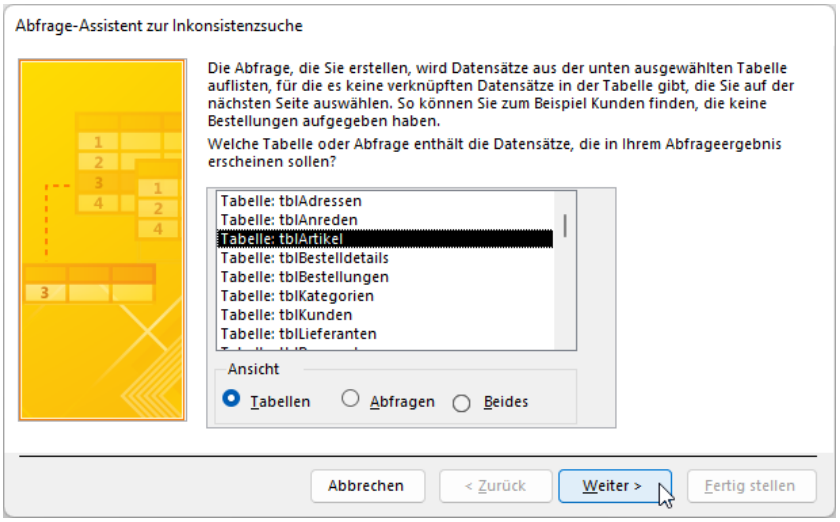


Bild 2: Der Dialog **Neue Abfrage**

Dies zeigt den Dialog **Neue Abfrage** an, wo wir neben ein paar weiteren Assistenten auch den Aufruf des Abfrage-Assistenten zur Inkonsistenzsuche finden (siehe Bild 2).

Wählen wir diesen Eintrag aus, erscheint auch gleich der erste Teil des Abfrage-Assistenten zur Inkonsistenzsuche (siehe Bild 3).

Hier wählen wir als Erstes die Tabelle aus, deren Datensätze in der Abfrage ermittelt werden sollen – wenn wir also die Kunden ermitteln wollten, die noch keine Bestellung aufgegeben haben, könnten wir das tun, aber wir wollen ein anderes Beispiel verwenden als im Assistenten angegeben.

Artikel ermitteln, die noch nie bestellt wurden

Also entscheiden wir uns dafür, die Artikel zu ermitteln, die noch nie erstellt wurden. Wir wählen im ersten Schritt also die Tabelle **tblArtikel** aus.

Im zweiten Schritt würden wir dann die Tabelle **tblBestelldetails** auswählen, weil dies die m:n-Verknüpfungstabelle ist, mit welcher die Artikel zu einer Bestellung hinzugefügt werden. Wir haben dann aber beim ersten Test gemerkt, dass es gar keine Artikel

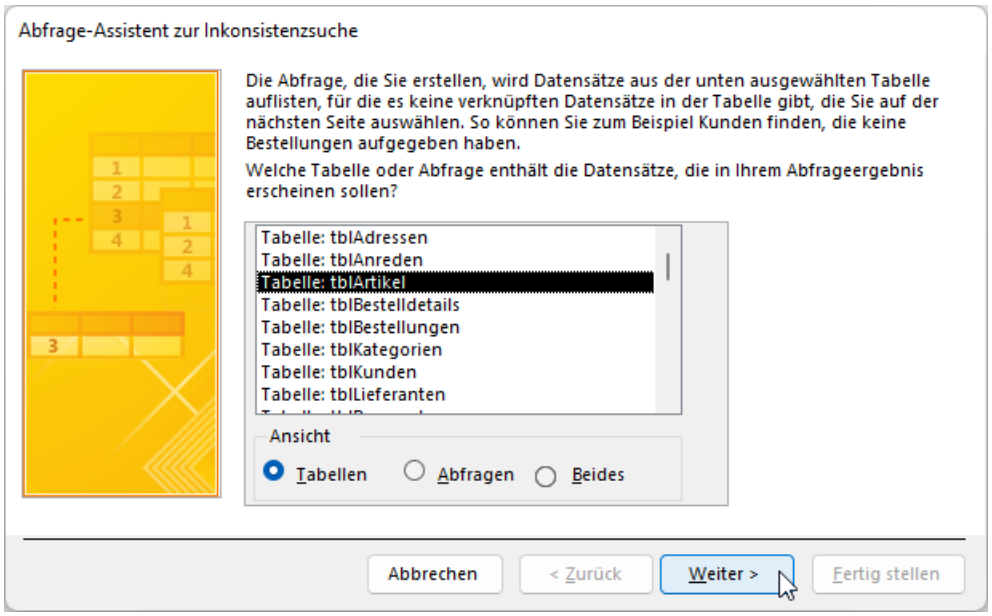


Bild 3: Schritt 1 des Assistenten

Abfragen [basics]: Unterabfragen

Unterabfragen? Was soll das nun wieder sein? Ich kenne Unterformulare, aber Unterabfragen? Nun: Unterabfragen sind ganz einfach Abfragen, deren Ergebnis als Kriterium in einer übergeordneten Abfrage verwendet wird oder deren Ergebnis als Teil des Ergebnisses der Hauptabfrage ausgegeben werden soll. Dabei gibt es für Unterabfragen spezielle Vorgaben, zum Beispiel dass diese nur ein einziges Feld zurückliefern dürfen (wobei das Feld auch eine Funktion eines Feldes sein kann wie eine Summe oder Anzahl). Wie genau man Unterabfragen definiert und wozu Du diese einsetzen kannst, zeige ich Dir in diesem Artikel.

Beispieldatenbank

Die Beispiele dieses Artikels findest Du in der Datenbank **2301_AbfragenBasics_Unterabfragen.accdb**.

Es geht nicht ohne SQL-Ausdrücke

Die schlechte Nachricht vorneweg: Im Gegensatz zu den übrigen Abfragen, die wir in der **Abfragen [basics]**-Beitragsreihe vorgestellt haben, kommen wir bei Unterabfragen nicht komplett mit dem Zusammenklicken von Abfragen über die Entwurfsansicht aus. Unterabfragen müssen nämlich als SQL-Ausdruck angegeben werden, also zum Beispiel wie folgt:

```
SELECT KundeID FROM tblKunden
```

Die gute Nachricht lautet jedoch: In den meisten Fällen können wir diese SELECT-Abfragen zuvor wiederum mit dem Abfrageeditor zusammenstellen und holen uns den benötigten SQL-Ausdruck dann aus der SQL-Ansicht der Abfrage.

SELECT ist Trumpf

Unterabfragen sind immer Auswahlabfragen, die allerdings eine entscheidende Einschränkung aufweisen: Sie enthalten nur ein Ergebnisfeld. Wir können allerdings auf mehrere Tabellen, Kriterien, Gruppierungen und weitere Techniken zurückgreifen.

Wo kann man Unterabfragen einsetzen?

Unterabfragen kann man vor allem an zwei Stellen nutzen:

- in Ausdrücken, also in berechneten Feldern
- in Kriterien von Abfragefeldern

Unterabfrage als Ausdruck

In einem berechneten Feld kann man das Ergebnis einer beliebigen Unterabfrage entweder als alleinigen Wert oder als Teil eines Ausdrucks mit anderen Informationen ausgeben. Wir könnten beispielsweise die Anzahl der Artikel der Tabelle in der gleichen Zeile mit den Daten des Artikels ausgeben. Das sieht zum Beispiel wie im Screenshot aus Bild 1 aus. Hier fügen wir also die folgende Abfrage als Unterabfrage hinzu:

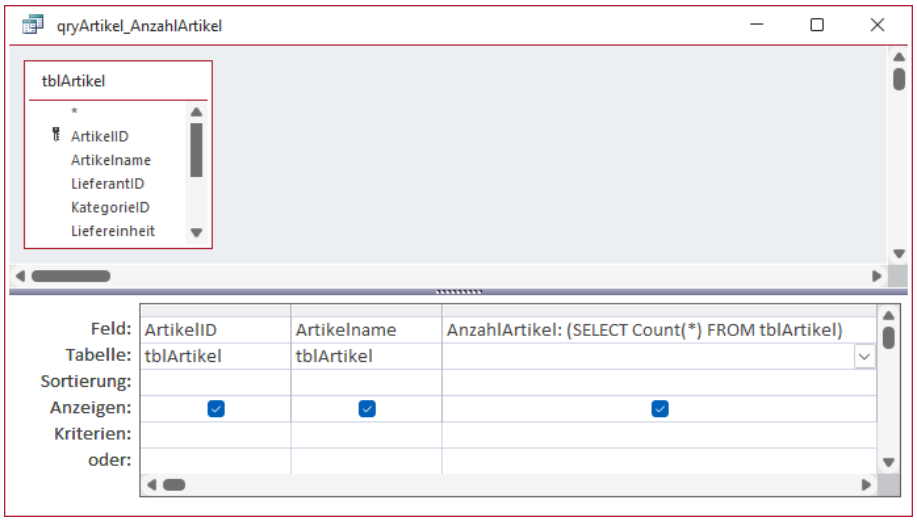


Bild 1: Beispiel einer Unterabfrage zur Ermittlung der Artikelanzahl

```
SELECT Count(*) FROM tblArtikel
```

Damit diese ihre Werte auch in dem von uns hinzugefügten Feld anzeigt, müssen wir einen Feldnamen für dieses berechnete Feld hinterlegen, in diesem Fall **AnzahlArtikel**. Außerdem ist es zwingend erforderlich, dass wir die Unterabfrage in Klammern einfassen, sodass dieser Ausdruck zum Einsatz kommt:

```
AnzahlArtikel: (SELECT Count(*) FROM tblArtikel)
```

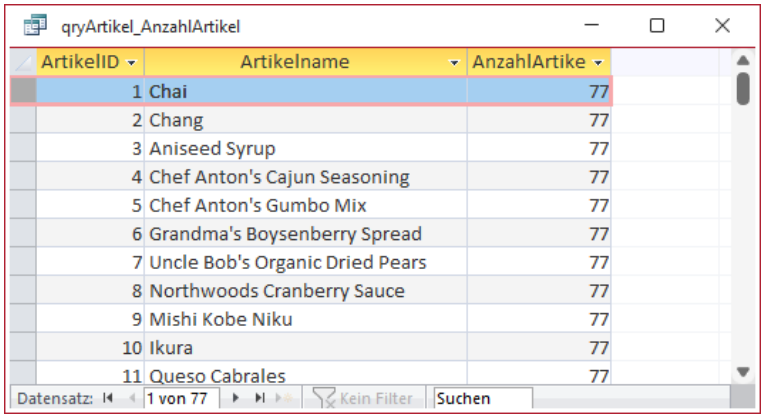


Bild 2: Die Anzahl der Artikel erscheint in jeder Zeile.

Wechseln wir in die Datenblattansicht, erhalten wir das Ergebnis aus Bild 2. Auf die gleiche Weise können wir auch Daten aus anderen Tabellen anzeigen.

Unterabfrage als Ausdruck mit Bezug zur Hauptabfrage

Das war ein sehr einfaches Beispiel, das kaum praktischen Nutzen hat. Was eher vorkommen könnte, wäre der Wunsch nach der Anzeige der Anzahl der Artikel der Kategorie des aktuellen Artikels. Sprich: Dies Hauptabfrage zeigt weiterhin alle Artikel der Tabelle **tblArtikel** an. In der Unterabfrage wollen wir die Anzahl der Artikel ermitteln, die der Kategorie angehören, zu welcher der aktuelle Artikel der Hauptabfrage gehört. Dazu müssen wir nun in der Unterabfrage ein Kriterium hinzufügen, dass sich auf die KategorieID des Artikels aus der Hauptabfrage bezieht.

Das ist nicht ganz so einfach, denn wie sollen wir bei der Formulierung des Kriteriums in der Unterabfrage das Feld **KategorieID** der Hauptabfrage mit dem gleichnamigen Feld der Unterabfrage vergleichen – wenn beide sich auf die gleiche Tabelle beziehen? Auch zu diesem Zweck gibt es in relationalen Datenbanken die Möglichkeit, einer Tabelle einen Aliasnamen zu geben. Wir legen also entweder in der Haupt- oder in der Unterabfrage fest, dass die Tabelle **tblArtikel** unter einem anderen Namen angesprochen werden soll, beispielsweise **t1**. Wir könnten auch für beide Instanzen der Tabelle **tblArtikel** in der Datenbank einen Aliasnamen vergeben, beispielsweise **t1** für die Tabelle in der Hauptabfrage und **t2**, damit es richtig klar wird.

Um für die Tabelle **tblArtikel** in der Hauptabfrage einen Aliasnamen festzulegen, aktivieren wir das Eigenschaftenblatt für diese Tabelle und legen den Aliasnamen für die Eigenschaft **Alias** fest (siehe Bild 3).

Allerdings erscheint nun überall in der Abfrage **t1** statt **tblArtikel**, was bei Verwendung mehrerer Tabellen in der Abfrage nicht zur Übersicht beiträgt (siehe Bild 4).

Daher würden wir in diesem Fall eher der Tabelle in der Unterabfrage den Aliasnamen **t1** zuweisen (zuvor ersetzen wir den Aliasnamen der Tabelle in der Hauptabfrage allerdings wieder durch den eigentlichen Namen der Tabelle, also **tblArtikel**). Um einer Tabelle im SQL-Code, den wir für Unterabfragen ange-

Bild 3: Festlegen eines Aliasnamens für eine Tabelle einer Abfrage

ben müssen, mit einem Aliasnamen zu versehen, geben wir hinter dem Tabellennamen im **FROM**-Bereich das **AS**-Schlüsselwort gefolgt vom Aliasnamen an:

```
SELECT Count(*) FROM tblArtikel AS t1
```

Dieser brauchen wir nun nur noch das Kriterium zuzuweisen, in dem wir die **KategorieID** der Tabelle der Unterabfrage mit der der Tabelle in der Hauptabfrage abgleichen:

```
SELECT Count(*) FROM tblArtikel AS t1
WHERE t1.KategorieID = tblArtikel.KategorieID
```

Diese Unterabfrage soll also die Anzahl der Datensätze aus der Tabelle **tblArtikel** holen, deren **KategorieID** mit der **KategorieID** des aktuellen Datensatzes in der Abfrage entspricht. Dazu fassen wir die Unterabfrage noch in Klammern ein und stellen dieser den Namen des berechneten Feldes voran. Der Abfrageentwurf sieht anschließend wie in Bild 5 aus.

Wechseln wir jetzt in die Datenblattansicht, erscheint zu jedem Artikel die Anzahl der Artikel der gleichen Kategorie (siehe Bild 6).

Anzahl verknüpfter Datensätze

Im nächsten Beispiel wollen wir herausfinden, wie viele verknüpfte Datensätze es für die Datensätze der zu untersuchenden Tabelle gibt. Im konkreten

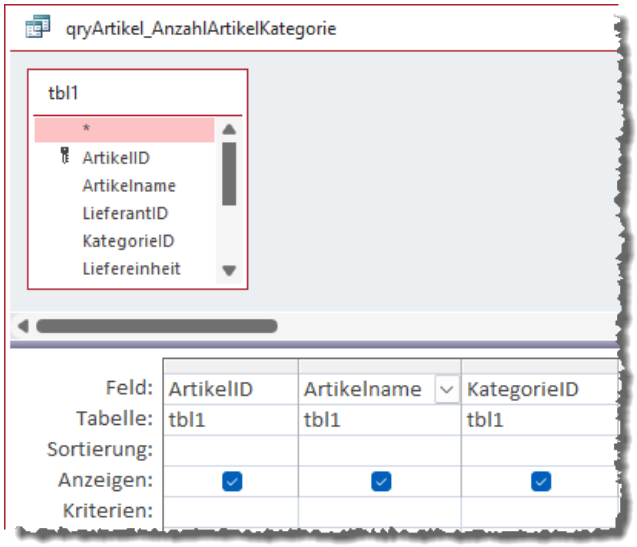


Bild 4: Tabelle mit Aliasname in einer Abfrage

Fall interessiert uns die Anzahl der Bestellungen eines jeden Kunden, die wir als zusätzliches Feld des Abfrageergebnisses ausgeben wollen. Dazu nutzen wir eine Abfrage, die in der Hauptabfrage Daten der Tabelle **tblKunden** anzeigt und in einem berechneten Feld namens **AnzahlBestellungen** die Anzahl der Datensätze der Tabelle **tblBestellungen**, die mit dem aktuellen Kundendatensatz verknüpft sind. Die dazu benötigte Unterabfrage enthält wieder eine Bedingung, welche ein Feld der Tabelle der Unterabfrage mit einem Feld der Tabelle der Hauptabfrage abgleicht. Diesmal handelt es sich allerdings um unterschiedliche Tabellen, daher brauchen wir keinen

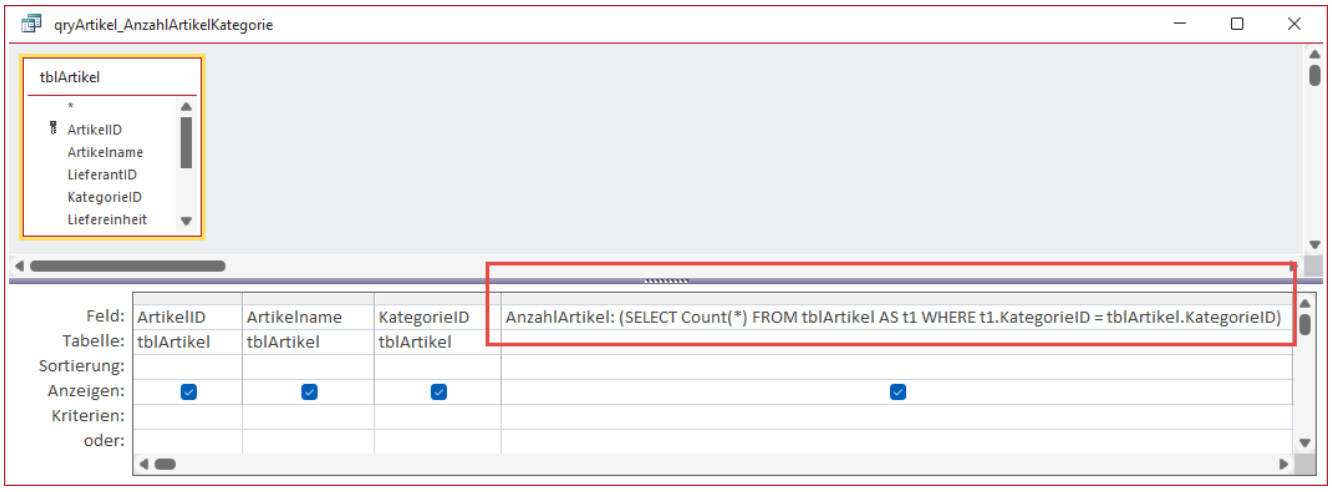


Bild 5: Ermitteln der Anzahl der Artikel der Kategorie des aktuellen Artikels

Formulare [basics]: Grundlagen

Formulare sind der Teil von Access, wo der Spaß beginnt. Während bei Tabellen die Planung und der Entwurf im Vordergrund stehen, weil alle übrigen Elemente darauf aufbauen und logischerweise angepasst werden müssen, wenn nachträglich Änderungen am Entwurf des Datenmodells stattfinden. Mit Formularen beginnt die Entwicklung des Teils der Anwendung, die der Benutzer zu sehen bekommt. Sie zeigen die Daten aus den Tabellen mit verschiedenen Steuerelementen an und verwenden andere Steuerelemente wie Schaltflächen, um verschiedene Aktionen durchzuführen. Dieser Artikel liefert einige Grundlagen für die Erstellung von Formularen.

Beispieldatenbank

Die Beispiele dieses Artikels findest Du in der Datenbank **FormulareBasics_Grundlagen.accdb**.

Anwendungsbereiche für Formulare

Formulare können für verschiedene Anwendungszwecke genutzt werden:

- Daten eingeben und bearbeiten: Access-Formulare bieten eine benutzerfreundliche Möglichkeit, um Daten in eine Datenbank einzugeben oder bestehende Daten zu bearbeiten.
- Daten filtern und sortieren: Formulare ermöglichen es Benutzern, Daten zu filtern und zu sortieren, um spezifische Informationen schnell zu finden.
- Berichte öffnen: Mit Access-Formularen kann man die Kriterien für die Daten vordefinieren, die in Berichten, die vom Formular aus geöffnet werden, angezeigt werden sollen.
- Daten visualisieren: Mit Access-Formularen können Benutzer Daten auf verschiedene Weise visualisieren, zum Beispiel durch Diagramme oder andere Steuerelemente wie das Webbrowser-Steuerelement.
- Datenvalidierung: Formulare können auch dazu genutzt werden, um sicherzustellen, dass Daten korrekt eingegeben werden, und entsprechende Meldungen anzeigen und das Speichern nicht valider Datensätze zu verhindern.
- Automatisierung von Abläufen: Access-Formulare können auch dazu verwendet werden, um komplexe Abläufe in der Datenbank automatisch zu steuern. Beispielsweise kann ein Formular entwickelt werden, das automatisch eine Rechnung erstellt, wenn ein Auftrag abgeschlossen wird, diesen als PDF-Datei speichert und sie dann durch Fernsteuerung von Outlook an den Kunden versendet.
- Benutzerschnittstelle: Formulare sind die Schnittstelle, die Benutzer überhaupt komfortabel mit den Daten einer Access-Datenbank interagieren lässt. Es gibt zwar sicher auch Datenbanken, in denen Benutzer nur mit Tabellen oder Abfragen arbeiten, um die Daten dort einzugeben. Eine benutzerfreundliche Formularoberfläche kann aber dazu beitragen, dass Benutzer die Datenbank effektiver nutzen und wertvolle Zeit sparen.

Formulare erstellen

Zum Erstellen von Formularen gibt es verschiedene Möglichkeiten. Diese finden wir im Ribbon im Tab **Erstellen** in der Gruppe **Formulare** (siehe Bild 1).

Hier finden wir die folgenden Befehle:

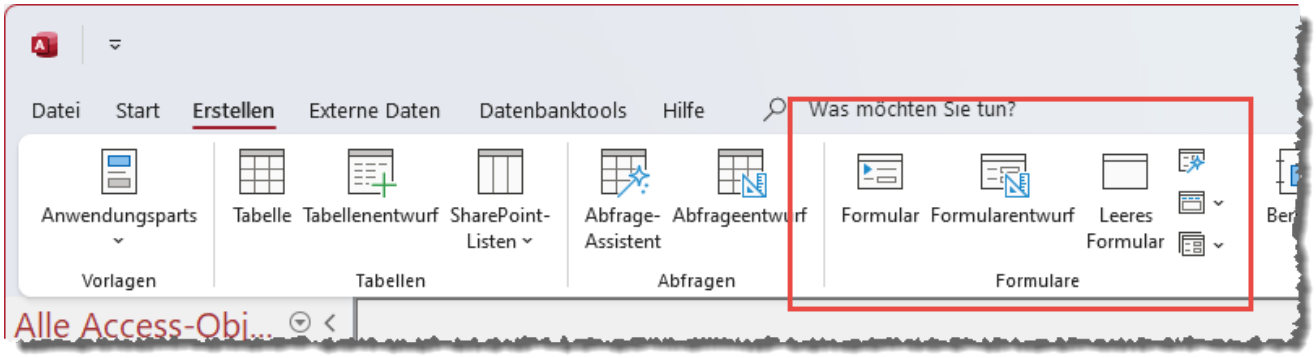


Bild 1: Erstellen von Formularen per Ribbon-Befehl

- **Formular**: Dieser Befehl erstellt ein neues Formular und verwendet dabei das aktuell im Navigationsbereich markierte Element als Datensatzquelle. Wenn dort beispielsweise eine Tabelle oder Abfrage markiert ist, zeigt das Formular Steuerelemente für die Felder dieser Tabelle oder Abfrage an. Wenn beispielsweise ein Formular markiert ist, wird ein neues Formular mit Daten basierend auf der Datenquelle des markierten Formulars erstellt. Wenn kein Element markiert ist, aus dem eine Datenquelle abgeleitet werden kann, ist auch der Befehl **Formular** nicht verfügbar.

- **Formularentwurf**: Öffnet ein neues, leeres Formular in der Entwurfsansicht. Dieses Formular hat die gängigen Standardeigenschaften und ist ansonsten ein unbeschriebenes Blatt – es ist weder an eine Datenquelle gebunden noch enthält es irgendwelche Steuerelemente.

- **Leeres Formular**: Öffnet prinzipiell das gleiche Formular wie der Befehl **Formularentwurf**, allerdings wird das Formular gleich in der Formularansicht angezeigt. Da man ein neues, leeres Formular typischerweise bearbeiten möchte, ist es sinnvoller, den vorherigen Befehl aufzurufen.

- **Formularassistent**: Dieser Befehl öffnet einen Dialog namens **Formular-Assistent**, mit dem wir ein neues Formular basierend auf einfachen Selektionen erstellen können – zum Beispiel durch Auswahl der Tabelle oder Abfrage, die als Datensatzquelle dienen soll und der anzuzeigenden Felder. Daneben legt man noch die grundlegende Strukturierung für das zu erstellende Formular fest, wobei **Einspaltig**, **Tabellarisch**, **Datenblatt** und **Blocksatz** zur Verfügung stehen.

- Navigationsformular mit verschiedenen Anordnungen von Registerkarten: Hiermit können wir Formulare erstellen, die initial mit einem Navigationssteuerelement ausgestattet werden. Die Verwendung von Navigationsformularen ist eher unüblich.

- **Weitere Formulare|Mehrere Elemente** (siehe Bild 2): Dies erstellt ein Formular in der sogenannten Endlosansicht. Dabei werden die Steuerelemente tabellarisch angeordnet, also die Beschriftungsfelder im Bereich **Formularkopf** und die eigentlichen Steuerelemente zur Anzeige der Daten im Detailbereich. Auch diese Vorlage verwendet die aktuell im Navigationsbereich markierte Tabelle oder Abfrage als Datensatzquelle.

- **Weitere Formularvorlagen|Datenblatt**: Dies erstellt ein Formular in der Datenblattansicht. Die Datenblattansicht ist eine automatisch erzeugte Ansicht, welche die im Formular enthaltenen Steuerelemente in der gleichen Ansicht anzeigt, die wir von Tabellen und Abfragen kennen.

- **Weitere Formularvorlagen|Geteiltes Formular**: Geteilte Formulare sind praktisch, wenn man

einerseits die Liste der Daten einer Tabelle sehen möchte und andererseits die Details des aktuell markierten Datensatzes entsprechend angeordnet darstellen will.

- **Weitere Formularvorlagen|Modales Dialogfeld**: Bei Verwendung dieser Vorlage sind einige Eigenschaften anders eingestellt als bei den übrigen Vorlagen. Diese sorgen dafür, dass verschiedene Elemente wie Datensatzmarkierer, Navigationsschaltflächen und Bildlaufleisten nicht angezeigt werden und das Formular im Access-Fenster zentriert geöffnet wird. Außerdem erhält es einen speziell aussehenden Rahmen und die Einstellung der Eigenschaft **Modal** auf den Wert **Ja** sorgt dafür, dass man erst mit der Arbeit mit den übrigen Elementen der Access-Benutzeroberfläche fortsetzen kann, wenn dieses Formular wieder geschlossen ist.

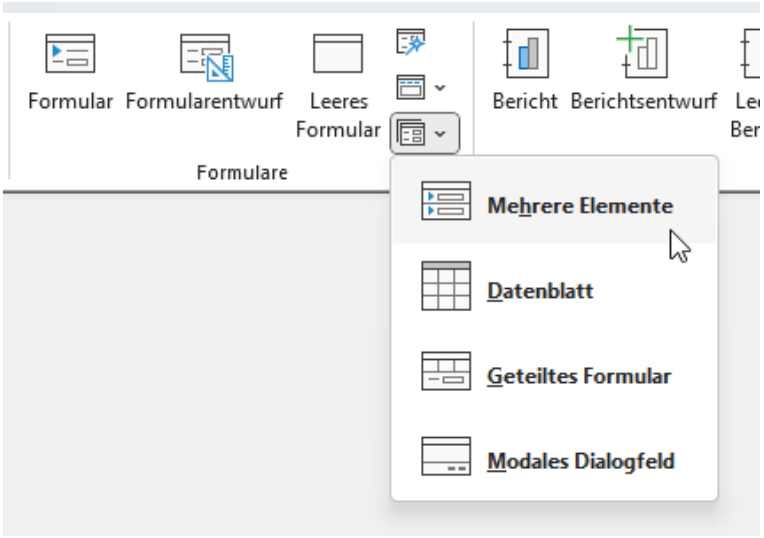


Bild 2: Weitere Formularvorlagen

Formularfunktionen schnell erklärt

Bevor wir uns die verschiedenen (sinnvollen) Möglichkeiten zum Erstellen von Formularen ansehen, wollen wir erst einmal ein einfaches Formular auf Basis einer Tabelle erstellen und uns anschauen, was wir damit überhaupt tun können.

Einfaches Formular auf Basis einer Tabelle anlegen

Dazu markieren wir die Tabelle **tblLaender** unserer Beispieldatenbank und betätigen den Ribbonbefehl **Erstellen|Formulare|Formular**. Einen Augenblick später sehen wir das Formular wie in Bild 3 vor uns. Dieses enthält im Kopfbereich eine Überschrift, welche dem Namen der Datensatzquelle entspricht, die im Formular dargestellt werden soll, hier also **tblLaender**. Darunter sehen wir drei Steuerelemente für die drei Felder der Tabelle. Für diese sind jeweils das Beschriftungsfeld links und das gebundene Steuerelement rechts angeordnet.

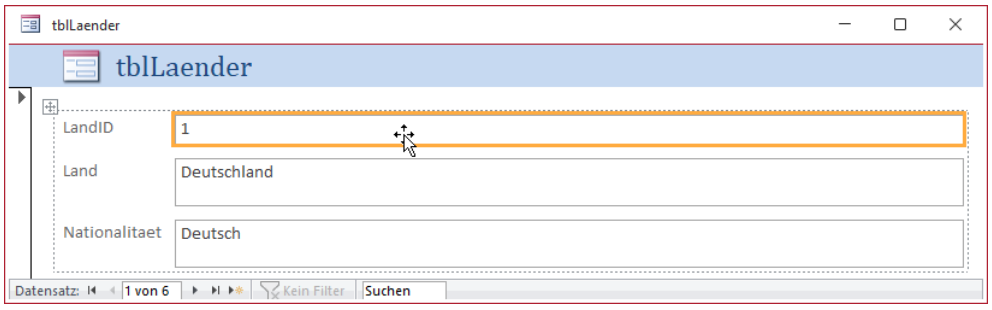


Bild 3: Ein neues Formular in der Layoutansicht

Formularansichten

Dieses Formular liegt uns in der Layoutansicht vor, was praktisch ist: In dieser Ansicht können wir nämlich einige Einstellungen des Layouts vornehmen, während wir gleichzeitig die Daten sehen.

Wir können die Daten allerdings in dieser Ansicht nicht bearbeiten, somit ist diese nicht für den Benutzer geeignet, sondern nur zur Verfeinerung und Optimierung des Entwurfs.

Damit ist diese Ansicht ein Hybrid zwischen zwei weiteren Ansichten, die wir noch im Detail kennenlernen werden:

- Die Entwurfsansicht zeigt den reinen Entwurf an, also die Anordnung der Steuerelemente ohne dass diese Daten anzeigen.

- Und die Formularansicht zeigt die Ansicht an, die der Benutzer später zu sehen bekommt – und an der er in dieser Ansicht keine Änderungen am Entwurf vornehmen kann.

Die Layoutansicht können wir beispielsweise dazu verwenden, die Breite der Steuerelemente so anzupassen, dass diese alle möglichen zu erwartenden Werte anzeigen können. Damit das in unserem Formular der Fall ist, markieren wir das Feld neben der Beschriftung **LandID** und ziehen den rechten Rand dieses Steuerelements so nach links, dass die Breite kleiner wird (siehe Bild 4). Wie wir hier sehen, ändern sich die Breiten der übrigen Steuerelemente ebenfalls. Das liegt daran, dass Access die Steuerelemente automatisch in ein Layout eingebettet hat, eine Art Raster, in dem bei Änderung der Breite einer Spalte direkt alle anderen Elemente dieser Spalte geändert werden – das Gleiche gilt für die Zeilenhöhen.

Wechseln der Ansicht

Um zwischen diesen Ansichten zu wechseln, gibt es mehrere Möglichkeiten. Die erste ist das Betätigen der rechten Maustaste auf dem Formular und die Auswahl eines der Einträge **Formularansicht**, **Layoutansicht** oder **Entwurfsansicht** des Kontextmenüs (siehe Bild 5).

Wechsel zur Formularansicht

Klicken wir hier auf Formularansicht, erhalten wir eine ähnliche Ansicht wie die Layoutansicht. Allerdings sind die orangen Markierungen und die gestrichelten Linien nicht mehr sichtbar, sondern nur noch die Bezeichnungsfelder und die Steuerelemente mit den Daten. Nun können wir die Daten auch bearbeiten (siehe Bild 6).

Bild 4: Anpassen von Steuerelementen in der Layoutansicht

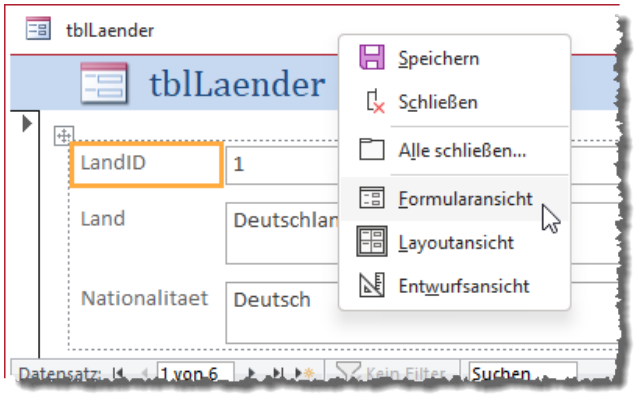


Bild 5: Wechseln der Ansicht eines Formulars

Bild 6: Die Formularansicht eines Formulars

Wechsel zur Entwurfsansicht

Für den Wechsel zur Entwurfsansicht verwenden wir die zweite Methode, nämlich einen Eintrag des Ribbon-Steuerelements **Start|Ansichten|Ansicht**. Hier finden wir ebenfalls die drei Einträge **Formular-**

Formulare [basics]: Beispiel Bücherverwaltung

Für die neue Artikelreihe **Formulare [basics]** wollen wir zuvor eine passende Beispieldatenbank erstellen. Mit dieser wollen wir Bücher verwalten. Dieser Artikel zeigt, welche Tabellen wir in einer Bücherverwaltung benötigen und welche Daten diese Tabellen aufnehmen. Wir zeigen, wie Du die Tabellen erstellst und wie diese Tabellen miteinander verknüpft sind. Wichtig ist, dass wir alle Verknüpfungstypen abdecken, damit wir Beispiele für die Darstellungsarten für die verschiedenen Konstellationen haben – und wir wollen auch alle Felddatentypen vorhalten, um alle möglichen Such- und Filtermöglichkeiten programmieren zu können.

Beispieldatenbank

Die Beispiele dieses Artikels findest Du in der Datenbank **FormulareBasics_BeispielBuecherverwaltung.accdb**.

Tabellen einer Bücherverwaltung

Für eine Bücherverwaltung benötigt man in der Regel folgende Tabellen:

- **Bücher**: Diese Tabelle enthält alle Informationen zu den Büchern wie Titel, Autor, ISBN-Nummer, Verlag, Erscheinungsjahr, Genre, Anzahl der Seiten, Erscheinungsland und so weiter.
- **Ausleihe**: Diese Tabelle enthält Informationen darüber, wer das Buch ausgeliehen hat, wann es ausgeliehen wurde und wann es zurückgegeben werden soll. Zusätzlich können noch Spalten für die Ausleihdauer und eventuelle Strafen hinzugefügt werden.
- **Leser**: Diese Tabelle enthält Informationen über die Kunden, die die Bücher ausleihen. Dazu gehören Name, Adresse, Telefonnummer und E-Mail-Adresse.
- **Autoren**: Diese Tabelle enthält Informationen über die Autoren, die die Bücher geschrieben haben. Dazu gehören Name, Geburtsdatum, Nationalität, Bibliographie und gegebenenfalls weitere Informationen.
- **Verlage**: Diese Tabelle enthält Informationen über die Verlage, bei denen die Bücher veröffentlicht wurden. Dazu gehören Name, Adresse, Telefonnummer, E-Mail-Adresse und weitere Informationen.
- **Kategorien**: Diese Tabelle enthält Informationen über die Kategorien, in die die Bücher eingeteilt werden können. Dazu gehören Name, Beschreibung und eventuell weitere Informationen (zum Beispiel Sachbuch, Belletristik et cetera).
- **Genre**: Genre des Buchs, also beispielsweise Krimi, Horror, Liebe, ...
- **Länder**: Länder für verschiedene Elemente wie Nationalität von Autoren oder Erscheinungsland von Büchern.
- Daneben gibt es noch Verknüpfungstabellen, die dafür sorgen, dass ein oder mehrere Datensätze der einen Tabelle mit einem oder mehreren Datensätzen der anderen Tabelle verknüpft werden können. Diese verknüpfen beispielsweise die Tabellen mit den Büchern und den Autoren und die mit den Büchern und Genres. Außerdem ist prinzipiell auch die Tabelle mit den Ausleihen eine solche Verknüpfungstabelle, wenn auch mit weiteren Daten als nur den reinen Verknüpfungsdaten. Mehr dazu weiter unten!

Damit haben wir genügend Material, mit denen wir die Formulare dieser Artikelreihe bestücken können. Wir haben Tabellen, die nicht über Fremdschlüsselfelder mit anderen Tabellen verknüpft sind und die wir in einem einfachen Formular abbilden können, das nur an diese Tabelle gebunden ist. Wir haben 1:n-Beziehungen dabei, wo Daten über ein Nachschlagefeld ausgewählt werden (Lookupbeziehung, Bücher und Genre) und wir haben 1:n-Beziehungen, wo wir die Daten über ein Unterformular zu denen im Hauptformular zuordnen können (beispielsweise Verlage und Bücher).

Und wir haben eine m:n-Beziehung zwischen den beiden Tabellen mit den Büchern und Kategorien, damit wir jedem Buch eine oder mehrere Kategorien zuweisen können, das Gleiche können wir noch mit Autoren und Büchern erledigen.

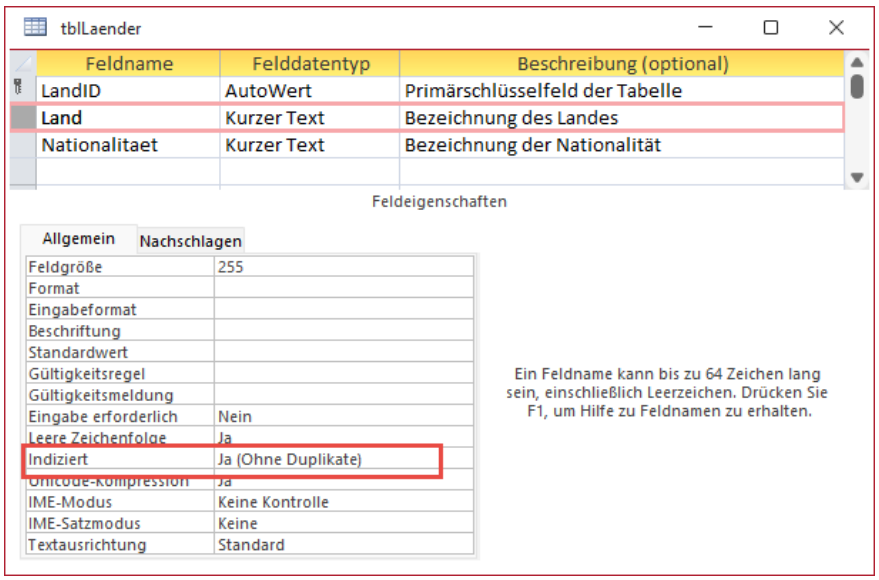


Bild 1: Tabelle zum Speichern der Länder

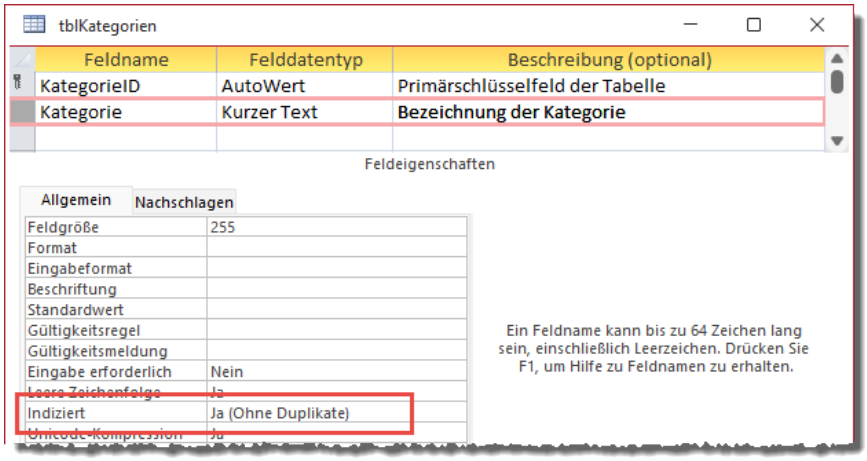


Bild 2: Tabelle zum Speichern der Kategorien

Reihenfolge beim Erstellen des Datenmodells

Wenn wir ein Datenmodell erstellen, starten wir am einfachsten mit den Tabellen, die keine Fremdschlüsselfelder zur Verknüpfung mit den Daten anderer Tabellen enthalten. Erst danach legen wir die Tabellen an, die über Fremdschlüsselfelder mit den zuerst angelegten Tabellen verknüpft sind. Der Hintergrund ist, dass wir dann die Fremdschlüsselfelder direkt mit dem Nachschlage-Assistenten hinzufügen können und nicht mit der Erstellung zweiter Tabellen gleichzeitig beschäftigt sind.

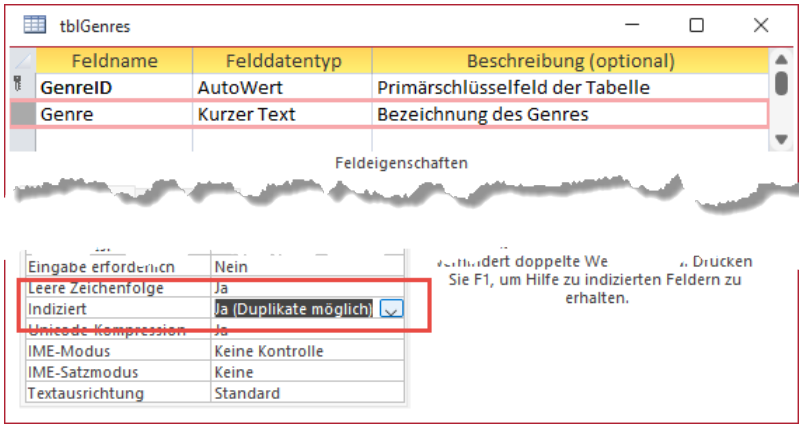


Bild 3: Tabelle zum Speichern der Genres

Am Besten beginnt man mit den sogenannten Lookup-Tabellen, die meist lediglich ein Primärschlüsselfeld sowie eine Bezeichnung enthalten. In diesem Fall finden wir mit der Tabelle **tblLaender** ein passendes Exemplar.

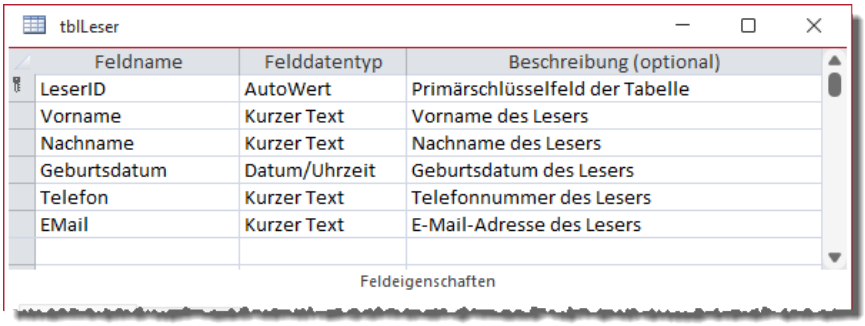


Bild 4: Tabelle zum Speichern der Leser

Die Tabelle tblLaender

Die Tabelle **tblLaender** enthält drei Felder: Das Primärschlüsselfeld **LandID**, das Feld mit der Bezeichnung des Landes namens **Land** sowie ein Feld namens Nationalität. Für das Primärschlüsselfeld legen wir den Datentyp **Autowert** fest. Für das Feld **Land** definieren wir einen eindeutigen Index, damit jedes Land nur einmal eingegeben werden kann (siehe Bild 1).

Diese Tabelle dient später als Nachschlagetabelle gleich für zwei weitere Tabellen. In der Tabelle **tblBuecher** wird es für die Auswahl des Erscheinungslands verwendet, in der Tabelle **tblAutoren** für die Nationalität des jeweiligen Autors.

Die Tabelle tblKategorien

Ähnlich aufgebaut wie die Tabelle **tblLaender** ist die Tabelle **tblKategorien**. Sie enthält ebenfalls nur ein Primärschlüsselfeld namens **KategorieID** sowie ein Feld zur Eingabe der jeweiligen Bezeichnung mit dem Datentyp **Kurzer Text**.

Auch für dieses Feld legen wir einen eindeutigen Index fest, damit jede Kategorie nur einmal vorkommt (siehe Bild 2).

Die Tabelle tblGenres

Und mit der Tabelle **tblGenres** erhalten wir die dritte Tabelle, die als Nachschlagetabelle dient (siehe Bild 3). Neben dem Primärschüsselfeld **GenreID** enthält die Tabelle das mit einem eindeutigen Index versehene Feld **Genre**.

Die Tabelle tblRollen

Mit dieser Tabelle wollen wir die verschiedenen Rollen festhalten, die Autoren eines Buchs einnehmen können – zum Beispiel Hauptautor, Co-Autor et cetera.

Die Tabelle tblLeser

Wir wollen in unserer Bücherverwaltung auch festhalten, wem wir ein Buch aus unserer Sammlung geliehen haben. Die entsprechende Tabelle heißt **tblLeser** und sieht in der Entwurfsansicht wie in Bild 4 aus.

Diese Tabelle enthält neben dem Primärschlüsselfeld **LeserID** die Textfelder **Vorname**, **Nachname**, **Telefon** und **EMail** sowie ein Datumsfeld namens **Geburtsdatum**.

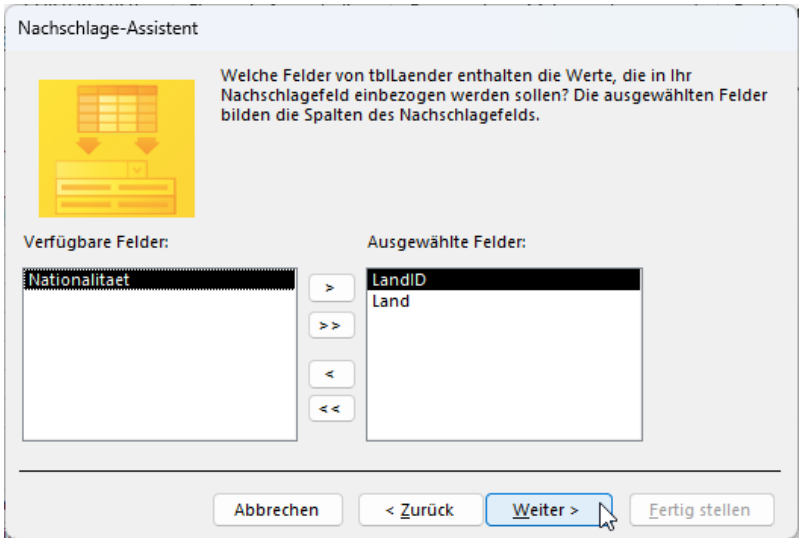


Bild 5: Hinzufügen der Felder für das Nachschlagefeld

Formulare [basics]: Doppelpunkt per Vorlage

Viele Programmierer fügen den Bezeichnungsfeldern von gebundenen Steuerelementen in Formularen Doppelpunkte hinzu. Wenn man dies für alle Bezeichnungsfelder in allen Formularen von Hand erledigen muss, ist das eine Menge Arbeit. Dies kann man vereinfachen, indem man eine Formularvorlage erstellt und darin für Bezeichnungsfelder eine nur dort verfügbare Eigenschaft einstellt. Wie das gelingt, zeigen wir in diesem Artikel.

Beispieldatenbank

Die Beispiele dieses Artikels findest Du in der Datenbank **FormulareBasics_Vorlagen.accdb**.

Ausgangssituation: Viel Handarbeit

Wer selbst einmal eine Reihe von gebundenen Formularen angelegt hat, kennt das: Man zieht die Steuerelemente aus der Feldliste in den Detailbereich der Entwurfsansicht und dort landen diese dann inklusive Bezeichnungsfelder. Dass die Bezeichnungsfelder dort angelegt werden, ist schon eine Erleichterung. Diese werden übrigens aus den Feldnamen übernommen. Manchmal muss man auch hier noch Änderungen vornehmen, beispielsweise bei Feldnamen wie **EMail** oder **AnredeID**. Aber auch das kann man vereinfachen, indem man im Tabellenentwurf die Eigenschaft Beschriftung des jeweiligen Feldes anpasst – in diesem Fall auf **E-Mail** oder **Anrede**. Man könnte theoretisch auch hier schon jeweils einen Doppelpunkt an die Beschriftung anhängen, damit man diese Aufgabe nur noch einmal je Feld erledigen muss und nicht mehr für jedes Formular, in dem das jeweilige Feld verwendet wird. Aber es geht noch besser!

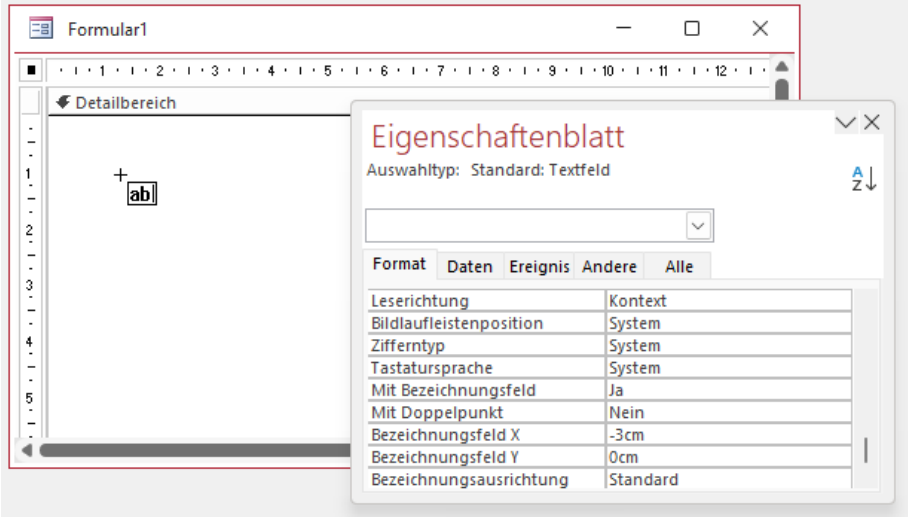


Bild 1: Anpassen der Standardeigenschaften eines Steuerelements

Doppelpunkt automatisch aktivieren

Wenn wir das Hinzufügen von Doppelpunkten für ein Formular aktivieren wollen, öffnen wir das Formular in der Entwurfsansicht. Dann beginnen wir so, als ob wir ein Steuerelement, beispielsweise ein Textfeld, zum Formular hinzufügen wollen. Dazu klicken wir dieses in der Sammlung der Steuerelemente im Ribbon an, aber wir fügen es nicht direkt im Formular ein. Stattdessen zeigen wir das Eigenschaftenblatt an (siehe Bild 1). Hier finden wir im unteren Bereich der Registerseite Format einige Eigenschaften, die wir von bereits angelegten Steuerelementen nicht kennen.

Für uns ist in diesem Fall die Eigenschaft **Mit Doppelpunkt** interessant. Wir stellen diese auf **Ja** ein. Anschließend können wir das Textfeld im Formular platzieren. Und siehe da: Das Bezeichnungsfeld enthält nun einen abschließenden Doppelpunkt (siehe Bild 2).

Fügen wir nun weitere Steuerelemente des Typs Textfeld hinzu, bleibt der Effekt erhalten. Für dieses

Formular ist die Eigenschaft **Mit Doppelpunkt** nun dauerhaft auf Ja eingestellt. Das funktioniert auch, wenn wir Felder aus der Feldliste in den Formularentwurf ziehen.

Eigenschaft für andere Steuerelemente einstellen

Allerdings landen beim Hineinziehen von Feldern aus der Feldliste nicht nur Textfelder im Formularentwurf. Wenn wir **Ja/Nein**-Felder oder Nachschlagefelder im Tabellenentwurf definiert haben, landen diese als Kontrollkästchen oder Kombinationsfeld im Formular. Damit auch diese mit dem Doppelpunkt ausgestattet werden, gehen wir genau wie beim Textfeld vor: Wir markieren das Steuerelement im Ribbon, so als ob wir es gleich zum Formular hinzufügen wollten, und stellen in den Eigenschaften **Mit Doppelpunkt** auf **Ja** ein.

Für alle Formulare

Nun haben wir diese Änderung zwar für ein Formular vorgenommen, aber die meisten Anwendungen verwenden mehr als nur ein Formular mit Daten aus einer Tabelle oder Abfrage. Wenn wir den Doppelpunkt in allen Formularen verwenden wollen, die wir über das Ribbon mit dem Befehl **Erstellen|Formulare|Formularentwurf** erstellen, können wir dafür eine Vorlage anlegen.

Dazu erstellen wir ein neues, leeres Formular. Für dieses durchlaufen wir die oben genannten Schritte für alle relevanten Steuerelemente, wobei es in der Regel ausreicht, Textfelder, Kontrollkästchen und Kombinationsfelder zu berücksichtigen. Es kann aber auch nicht schaden, weitere Steuerelement wie das Listenfeld oder das Unterformular mit Doppelpunkten für die Beschriftung zu versehen.

Diesmal fügen wir allerdings keines der Steuerelemente tatsächlich zum Formular hinzu, sondern

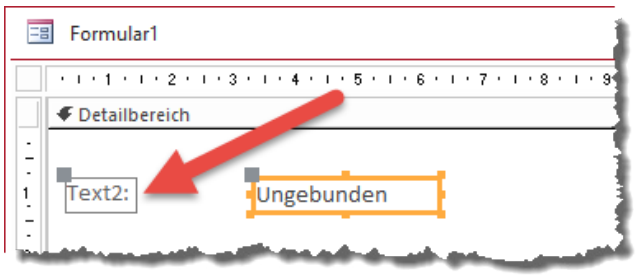


Bild 2: Bezeichnungsfeld mit Doppelpunkt

wir speichern dieses unter dem Namen **Normal**. Dies ist der Standardname für eine Formularvorlage.

Wenn wir nach dem Speichern dieser Vorlage ein neues Formular mit **Erstellen|Formulare|Formularentwurf** anlegen und diesem beispielsweise die Felder einer als Datensatzquelle angegebenen Tabelle aus der Feldliste hinzufügen, werden die Beschriftungsfelder der entsprechenden Steuerelemente fortan mit Doppelpunkten ausgestattet.

Formularvorlage einstellen

Sollte das einmal nicht funktionieren, kann es daran liegen, dass für die zu verwendende Formularvorlage ein anderer Name als **Normal** eingestellt ist. Das prüfen wir, indem wir die Access-Optionen öffnen und hier zum Bereich **Objekt-Designer** wechseln.

Hier finden wir unter **Entwurfsansicht für Formulare/Berichte** die Eigenschaft **Formularvorlage** (siehe Bild 3). Diese sollte den Wert **Normal** enthalten.

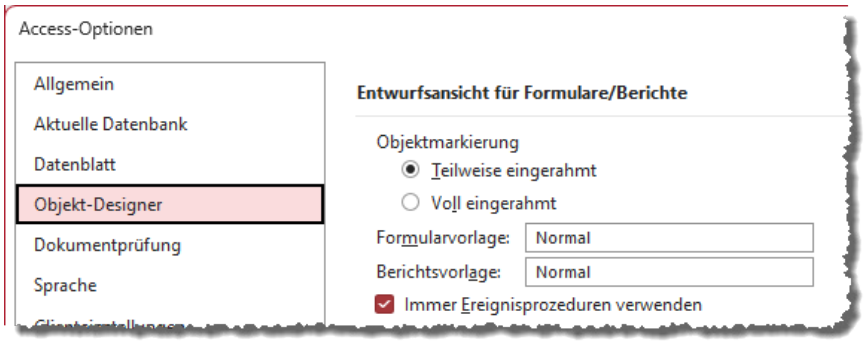


Bild 3: Name der Formularvorlage

Formulare [basics]: Steuerelemente ausrichten

Steuerelemente zum Formularentwurf ist eine Sache, das optisch ansprechende Anordnen eine andere. Dieser Artikel klärt die technischen Möglichkeiten, mit denen wir die Ausrichtung und die Größe von Steuerelementen einstellen können und mit welchen Tricks wir schnellere Ergebnisse erhalten können. Dabei nutzen wir sowohl die manuelle Ausrichtung über das Verschieben mit Maus und Tastatur, aber auch Ribbon- und Kontextmenübefehle zum Anpassen von Position und Größe. Eine wichtige Rolle spielt schließlich noch das Entwurfsraster.

Beispieldatenbank

Die Beispiele dieses Artikels findest Du in der Datenbank **FormulareBasics_SteuerelementeAusrichten.accdb**.

Beispielformular

Als Beispiel verwenden wir das einfache Formular namens **frmLeserDetails**, das wir im Artikel **Formulare [basics]: Daten aus einer Tabelle** (**www.access-basics.de/621**) anlegen (siehe Bild 1).

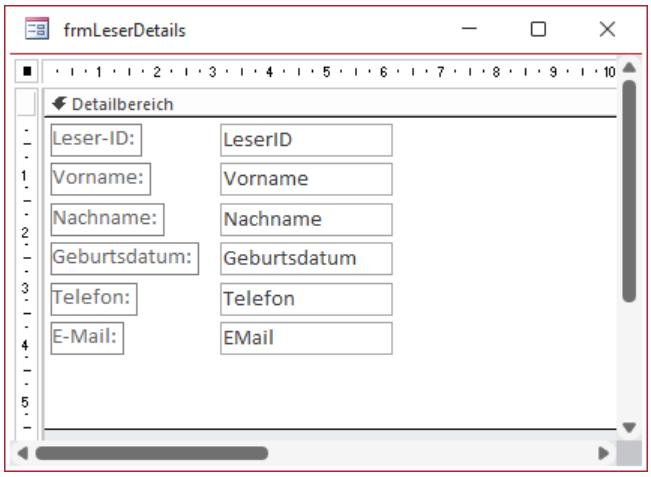


Bild 1: Formular-Entwurf unseres Beispielformulars

Steuerelemente markieren

Die nachfolgend beschriebenen Techniken funktionieren meist nicht nur mit einem Steuerelement, sondern auch mit mehreren Steuerelementen gleichzeitig.

Dabei sind die Techniken zum Markieren mehrerer Steuerelemente hilfreich. Am einfachsten ist es, mit der Maus einen Rahmen um alle zu bearbeitenden Steuerelemente zu ziehen. Die markierten Steuerelemente werden dann farbig hervorgehoben.

Manchmal funktioniert das jedoch nicht, weil sich mittendrin Steuerelemente befinden, die nicht angepasst werden sollen. In diesem Fall nehmen wir zusätzlich die Tastatur zu Hilfe.

Wir können sowohl die **Strg**- als auch die **Umschalt**-Taste gedrückt halten und dabei Steuerelemente zur Auswahl hinzufügen oder daraus entfernen.

Wenn man schnell alle Steuerelemente außer einem oder einigen wenigen markieren möchte, kann man die beiden Techniken kombinieren: Wir ziehen dann zuerst einen Rahmen um alle Steuerelemente auf und entfernen dann die Markierung der nicht zu markierenden Steuerelemente, indem wir diese bei gedrückter **Strg**- oder **Umschalt**-Taste anklicken.

Position und Größe per Maus

Die intuitivste Möglichkeit, Position und Größe von Steuerelementen zu ändern, ist das Anklicken mit der Maus und das anschließende Bearbeiten. Platziert man die Maus auf dem Markierungsrahmen, kann man das Steuerelement durch Ziehen verschieben (siehe Bild 2).

Platzieren wir den Zeiger hingegen auf einem der Eckpunkte, können wir die Größe ändern. Die Punkte in der Mitte der Linien dienen zum Anpassen der Höhe und Breite des Steuerelements (siehe Bild 3).

Steuerelement mit Beschriftungsfeld

Beim Ändern der Position eines Steuerelements mit einem Bezeichnungsfeld mit der Maus wird das Bezeichnungsfeld in die gleiche Richtung verschoben.

Wenn das Bezeichnungsfeld seine Position beim Verschieben des Steuerelements beibehalten soll, müssen wir das Steuerelement mit dem Mauszeiger an dem grauen Rechteck links oben anfassen und dann verschieben.

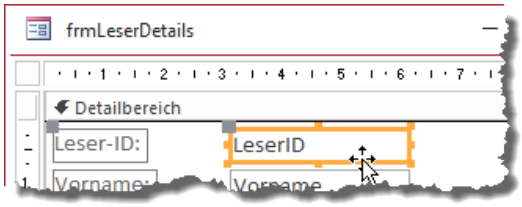


Bild 2: Verschieben eines Steuerelements

Das Gleiche gilt, wenn wir nur das Bezeichnungsfeld verschieben wollen.

Wenn wir eines von beiden Elementen verschieben kann, es schnell passieren, dass wir nicht nur die horizontale Position ändern (was normalerweise gewünscht ist), sondern auch die vertikale Position.

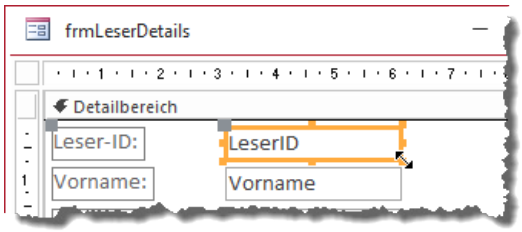


Bild 3: Ändern der Größe eines Steuerelements

In diesem Fall halten wir beim Verschieben die Umschalttaste gedrückt. Access erkennt die initiale Richtung beim Verschieben und erlaubt kein Verschieben in die andere Richtung (also keine vertikale Verschiebung, wenn wir horizontal verschieben und umgekehrt).

Position per Tastatur

Wenn wir die Position eines Steuerelements mit der Tastatur verändern wollen, markieren wir das Steuerelement zunächst. Dann können wir es pixelweise oder in größeren Schritten bewegen, indem wir die **Nach links**-, **Nach oben**-, **Nach rechts**- oder **Nach unten**-Tasten verwenden:

- Wenn wir die **Strg**-Taste beim Betätigen der Tasten gedrückt halten, verschieben wir das Steuerelement pixelweise.
- Wenn wir die Tasten ohne gedrückte **Strg**-Taste betätigen, verschieben wir das Steuerelement in größeren Schritten.

Was heißt nun »in größeren Schritten«? Das erläutern wir weiter unten unter **Das Entwurfsraster**.

Ein getrenntes Verschieben von zusammengehörenden Bezeichnungsfeldern und Steuerelementen gelingt mit der Tastatur nicht.

Größe per Tastatur anpassen

Um die Größe von Steuerelementen mit der Tastatur zu ändern, markieren wir wieder das zu ändernde Steuerelement (oder auch mehrere Steuerelemente gleichzeitig).

Genau wie die Position können wir auch die Größe pixelweise oder in größeren Schritten verändern:

- Pixelgenau Breite vergrößern: **Strg + Umschalt + Nach Rechts**
- Pixelgenau Breite verkleinern: **Strg + Umschalt + Nach Links**
- Pixelgenau Höhe vergrößern: **Strg + Umschalt + Nach unten**
- Pixelgenau Höhe verkleinern: **Strg + Umschalt + Nach oben**

Um die Größe in größeren Schritten zu ändern, verwenden wir die oben genannten Tastenkombinationen ohne **Strg**-Taste, also zum Vergrößern der Breite beispielsweise die Tastenkombination **Umschalt + Nach rechts**.