

## Formulare [basics]: m:n-Beziehung mit Unterformular

Tabellen in einer m:n-Beziehung sind Tabellen, die über eine Verknüpfungstabelle miteinander verknüpft sind, sodass jeder Datensatz der ersten Tabelle mit jedem Datensatz der zweiten Tabelle verknüpft werden kann. Die Verknüpfungstabelle realisiert das mit je einem Fremdschlüsselfeld zum Verknüpfen dieser beiden Tabellen. Wollen wir dies in Formularen darstellen, sodass der Benutzer komfortabel mit den Daten in dieser Beziehungsart arbeiten kann, können wir die Daten in einem Hauptformular mit einem Unterformular darstellen. Auch unter den m:n-Beziehungen gibt es verschiedene Variationen. Wir bilden in diesem Artikel die Variante ab, bei der die m:n-Verknüpfungstabelle neben den eigentlichen Verknüpfungsdaten noch weitere Informationen aufnimmt, die sich auf die jeweilige Kombination der Datensätze der beiden Tabellen beziehen.

### Beispieldatenbank

Die Beispiele dieses Artikels findest Du in der Datenbank **Formulare-Basics\_mnBeziehungenMitUnterformular.accdb**.

### Beispieltabellen der Bücherverwaltung

In unserer Bücherverwaltung finden wir eine geeignete Beziehung. Dabei handelt es sich um die Beziehung zwischen der Tabelle **tblBuecher** und der Tabelle **tblLeser**.

Aber in welchem Zusammenhang führen wir diese beiden Informationen in einer m:n-Beziehung zusammen?

Wir wollen unsere Bücherverwaltung auch dazu nutzen, um Bücher zu verleihen und dabei die Leihvorgänge zu dokumentieren. Deshalb wollen wir aufnehmen, welches Buch wir wem geliehen haben – und vor allem, wann es verliehen wurde und wann wir es wiederbekommen haben.

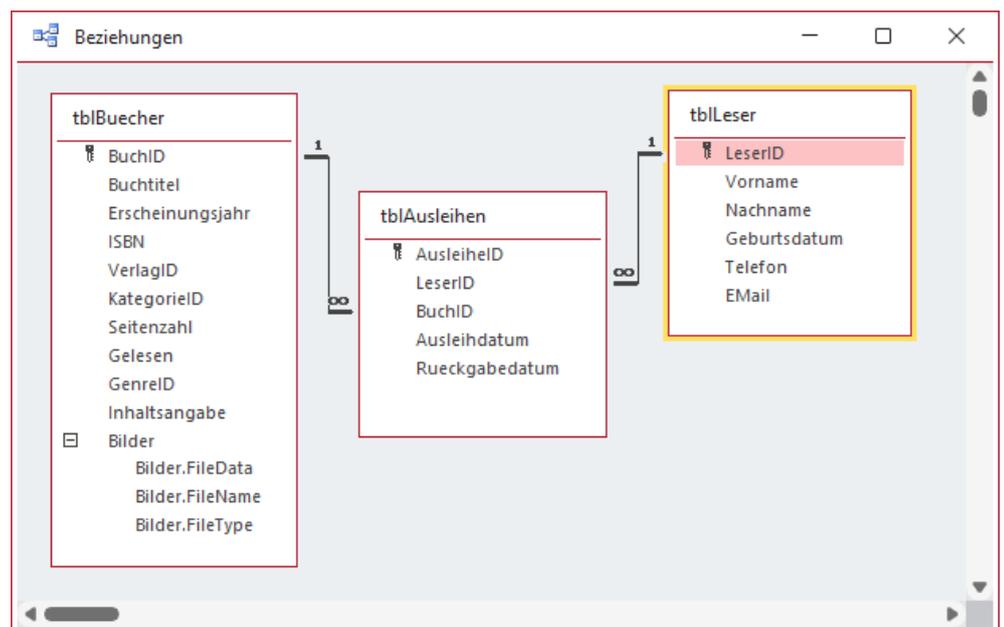


Bild 1: Beziehung zwischen Büchern und Lesern

Bild 1 zeigt die an der Beziehung beteiligten Tabellen, in diesem Fall **tblBuecher**, **tblLeser** und die Verknüpfungstabelle **tblAusleihen**. Die Tabelle **tblAusleihen** enthält neben dem Primärschlüsselfeld **AusleiheID** und den beiden Fremdschlüsselfeldern **LeserID** und **BuchID** noch zwei Felder namens **Ausleihdatum** und **Rueckgabedatum**, mit denen wir die Ausleihvorgänge dokumentieren können. Diese Tabellen wollen wir nun in Formularen bearbeiten können.

## Daten aus einer m:n-Beziehung in Formularen anzeigen

Dabei stellt sich erst einmal die Frage, wie wir die Daten aus diesen drei Tabellen in einem Formular anzeigen können.

Dazu schauen wir uns erst einmal die Verknüpfungstabelle für die m:n-Beziehung an (siehe Bild 2). Da wir, wie fast immer, die Fremdschlüsselfelder als Nachschlagfelder in den Tabellen definiert haben, können wir hier recht komfortabel jeweils einen Leser und ein Buch auswählen und das Ausleihdatum und gegebenenfalls auch das Rückgabedatum in dieser Tabelle einfügen. Was wollen wir denn noch mehr? Müssen wir überhaupt ein Formular erstellen, wenn wir die gewünschte Daten so komfortabel bereits über die Datenblattansicht der Tabelle eingeben können?

Davon abgesehen, dass wir, wenn wir diese Daten in einem Formular abbilden, noch wesentlich bessere Möglichkeiten zum Steuern und Verarbeiten der Eingaben hinzufügen können, wollen wir den Benutzer unserer Anwendungen grundsätzlich nicht direkt in den Tabellen der Datenbank arbeiten lassen.

Wenn wir nun also ein Formular erstellen, das genau diese Daten anzeigt und uns ihre Bearbeitung erlaubt, sind wir aber dennoch schon fast am Ziel, oder?

Ausleihe-ID	Leser	Buch	Ausleihdatum	Rückgabedatum
1	Stratmann, Adi	Anwendungen entwickeln mit Access	05.08.2024	
2	Birk, Wernfried	Access und SQL Server	04.08.2024	
3	Birk, Wernfried	Anwendungen entwickeln mit Access	03.08.2024	
4	Birk, Wernfried	Access und Office	02.08.2024	05.08.2024
(Neu)				

Bild 2: Informationen zu einer Ausleihe zusammentragen

## Formular zum Verwalten von Ausleihen

Wir lassen die Frage vorerst unbeantwortet und legen zuerst einmal ein solches Formular an. Dazu nutzen wir den Ribbonbefehl **Erstellen|Formulare|Formularentwurf** und erhalten ein neues, leeres Formular in der Entwurfsansicht. Diese weisen wir nun als Erstes die Tabelle `tblAusleihen` für die Eigenschaft **Datensatzquelle** hinzu (siehe Bild 3).

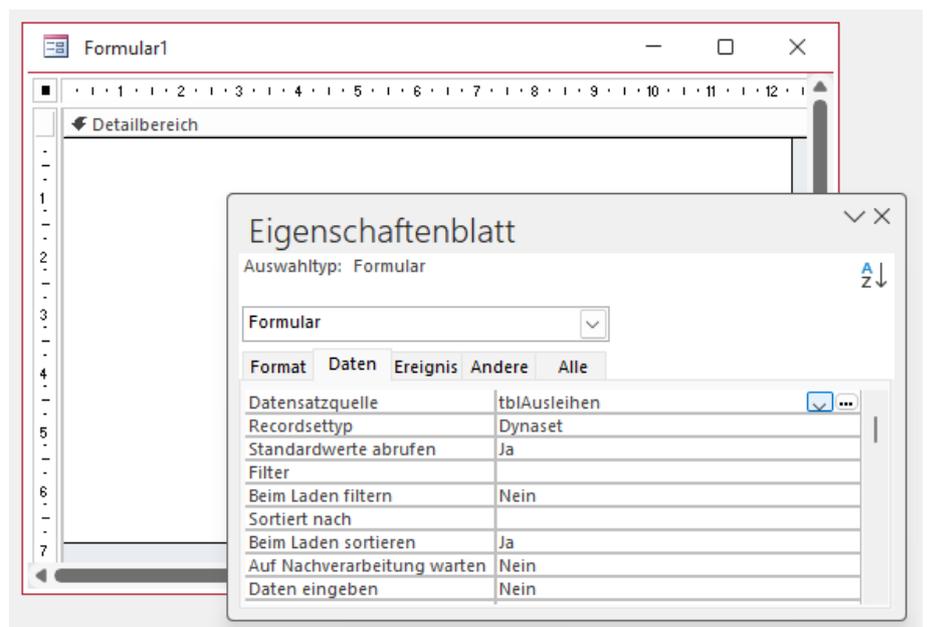


Bild 3: Hinzufügen der Datensatzquelle

Nun aktivieren wir mit dem Ribbonbefehl **Formularentwurf** | **Tools** | **Vorhandene Felder hinzufügen** die Anzeige der Feldliste der Datensatzquelle und ziehen zunächst einmal alle Felder aus dieser Liste in den Detailbereich des Formularentwurfs (siehe Bild 4).

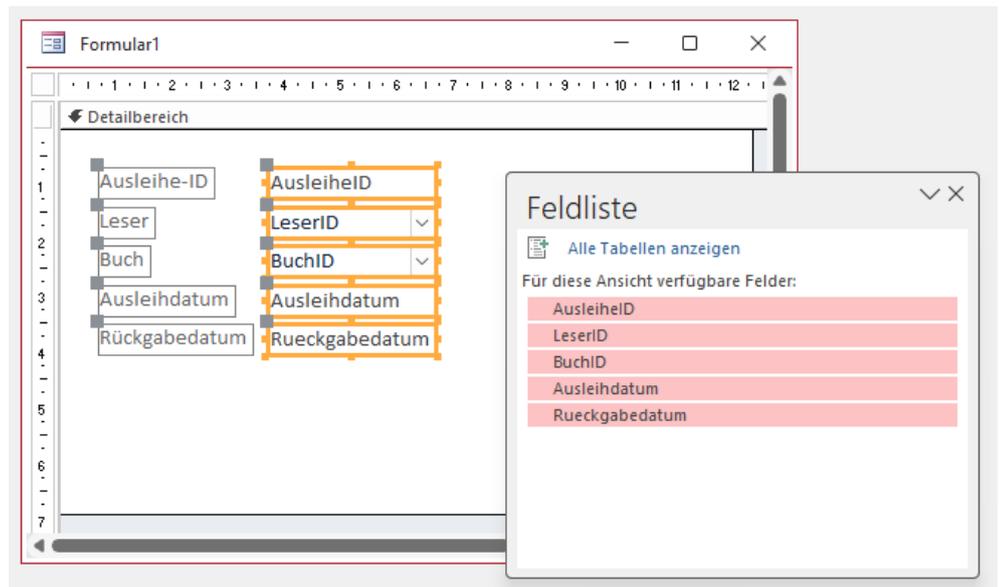


Bild 4: Hinzufügen der Felder der Tabelle tblAusleihen

Die Anordnung der Felder spielt erst einmal keine Rolle, da wir die Felder ohnehin in der Datenblattansicht anzeigen wollen. Dazu wechseln wir noch einmal von der Feldliste zum **Eigenschaftentblatt** und stellen hier die Eigenschaft **Standardansicht** auf **Datenblatt** ein. Um dies zu erledigen, müssen wir zunächst das Formular selbst markieren, damit seine Eigenschaften erscheinen. Dies erreichen wir, indem wir entweder oben links auf das kleine Kästchen der auf den leeren, grauen Bereich im Formular klicken oder indem wir im Eigenschaftentblatt den Eintrag **Formular** auswählen (siehe Bild 5).

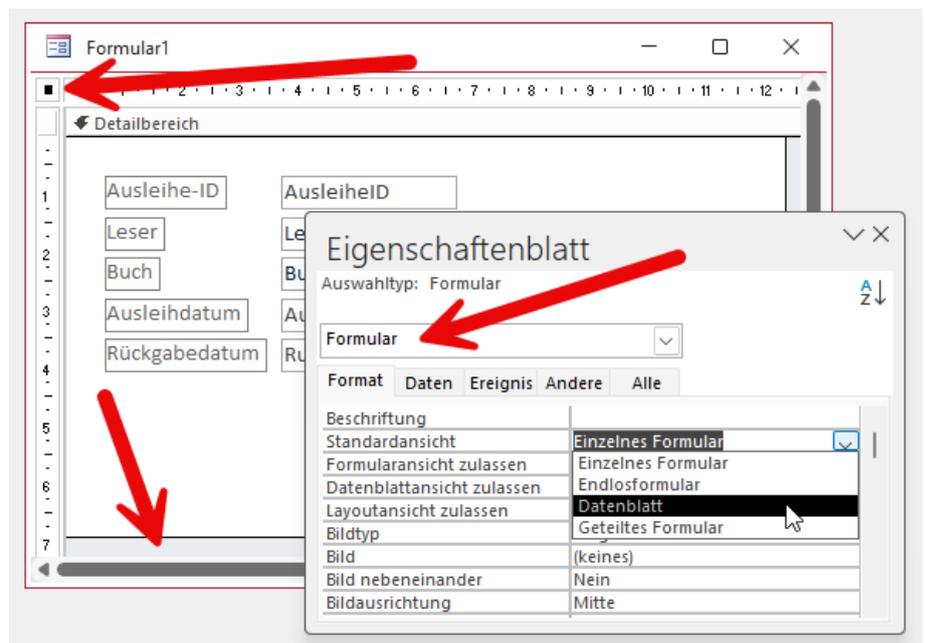


Bild 5: Einstellen der Standardansicht

Damit sind die Arbeiten an diesem Formular vorerst erledigt.

Wir speichern es unter der Bezeichnung **sfmAusleihen**.

Warum nutzen wir nicht **frmAusleihen**, sondern verwenden das Präfix, das normalerweise für Unterformulare eingesetzt wird? Das erfahren wir weiter unten.

Zunächst einmal wechseln wir nun in die Datenblattansicht. Das Formular liefert nun genau die gleiche Darstellung wie die Tabelle **tblAusleihen** in der Datenblattansicht (siehe Bild 6). Sind wir damit bereits fertig? Nein, jetzt geht der Spaß erst los, damit die Dateneingabe richtig komfortabel wird.

Wir können nun zwar bereits alle notwendigen Daten eingeben, aber dafür sehen wir auch immer alle Datensätze gleichzeitig – für alle Bücher und für alle Leser. Wir könnten zwar mit einem Klick auf den nach unten zeigenden Pfeil der Spalten Leser oder Buch ein Popup-Menü anzeigen, mit dem wir nach einem oder mehreren Lesern oder Büchern filtern können, aber so richtig komfortabel ist das nicht (siehe Bild 7). Viel schöner wäre es, wenn wir beispielsweise die Leser durchblättern könnten und dabei jeweils die entliehenen Bücher dieses Lesers sehen könnten.

Ausleihe-ID	Leser	Buch	Ausleihdatum	Rückgabedatum
1	Stratmann, Adi	Anwendungen entwickeln mit Access	05.08.2024	
2	Birk, Wernfried	Access und SQL Server	04.08.2024	
3	Birk, Wernfried	Anwendungen entwickeln mit Access	03.08.2024	
4	Birk, Wernfried	Access und Office	02.08.2024	05.08.2024
*	(Neu)			

Bild 6: Die Ausleih-Informationen in einem Formular in der Datenblattansicht

### Hauptformular für die m-Seite hinzufügen

Wir möchten dazu ein Hauptformular erstellen, das nur die Leser aus der Tabelle `tblLeser` anzeigt, und zwar jeweils nur einen. Dazu legen wir ein neues Formular an und weisen diesem als Datensatzquelle die Tabelle `tblLeser` hinzu. Diesmal lassen wir ein wenig mehr Sorgfalt bei der Anordnung der Steuerelemente aus der Tabelle `tblLeser` walten, denn das Formular soll nicht in der Datenblattansicht erscheinen, sondern in der Formularansicht, und nur einen Datensatz der Tabelle `tblLeser` gleichzeitig anzeigen. Das Ergebnis sieht etwa wie in Bild 8 aus.

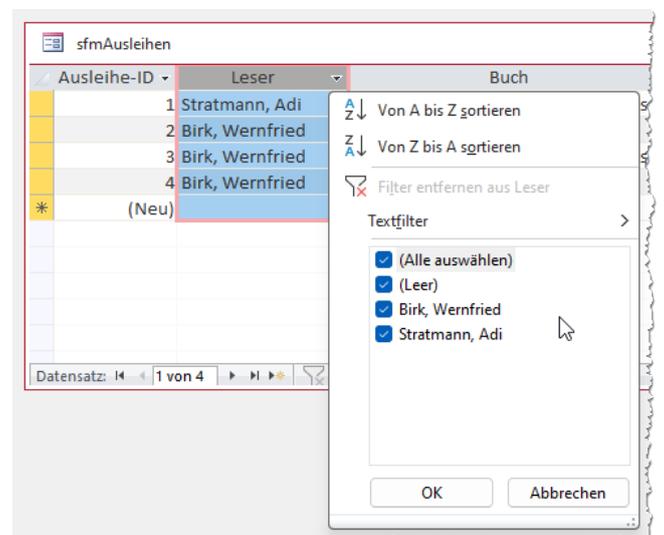


Bild 7: Filtermöglichkeit für die Leser

Wenn wir danach in die Formularansicht wechseln, sehen wir den ersten Datensatz der Datensatzquelle `tblLeser` im Formular `frmAusleihen` (siehe Bild 9).

### Haupt- und Unterformular zusammenführen

Damit kommen wir dem Ziel näher. Wir haben nun ein Formular, das

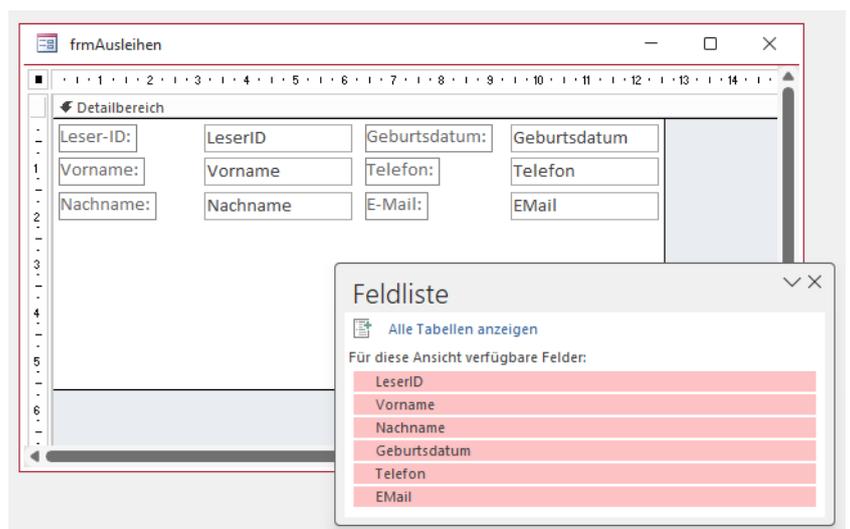


Bild 8: Hinzufügen der Felder zum Formular `frmAusleihen`

jeweils einen Leser gleichzeitig anzeigt und eines, das alle Ausleihen liefert. Wir bringen wir diese beiden so zusammen, dass diese sowohl den Leser als auch die von ihm ausgeliehenen Bücher anzeigt?

Dazu wollen wir nun die Liste der Ausleihen so in das Formular mit den Lesern integrieren, dass dieses genau die zu dem aktuell angezeigten Leser gehörenden Ausleihen anzeigt.

Und genau dafür ist das Steuerelement **Unterformular** zuständig. Wir haben mehrere Möglichkeiten, ein Unterformular zu einem Hauptformular hinzuzufügen und es mit dem darin anzuzeigenden Formular zu füllen. Wir schauen uns zunächst den umständlicheren Weg an, damit Du genau verstehst, welche Schritte Du hier einsparst. Danach zeige ich Dir, wie es auf die einfache Weise gelingt.

**Unterformular manuell einfügen**  
Voraussetzung für die nächste Schritte ist, dass das Hauptformular in der Entwurfsansicht angezeigt wird.

Als Erstes fügen wir ein Unterformular-Steuerelement aus dem Ribbon zu Formular hinzu. Dazu klicken wir entsprechende Element aus Bild 10 an.

Anschließend ziehen wir einen Rahmen von der Größe auf, die das Unterformular gleich einnehmen soll (siehe Bild 11).

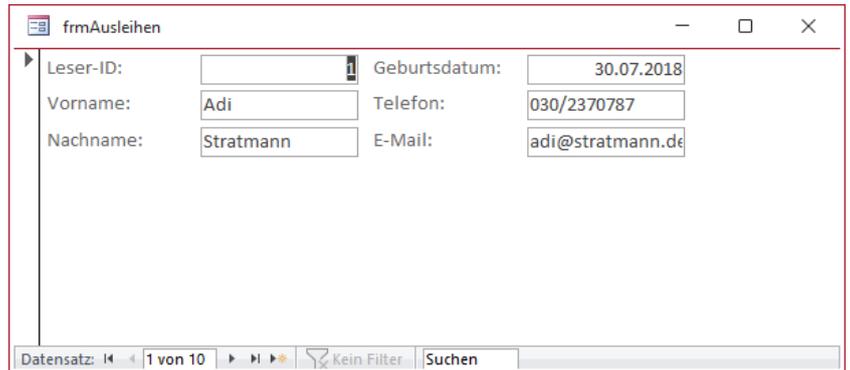


Bild 9: Das Hauptformular mit den Leserdaten

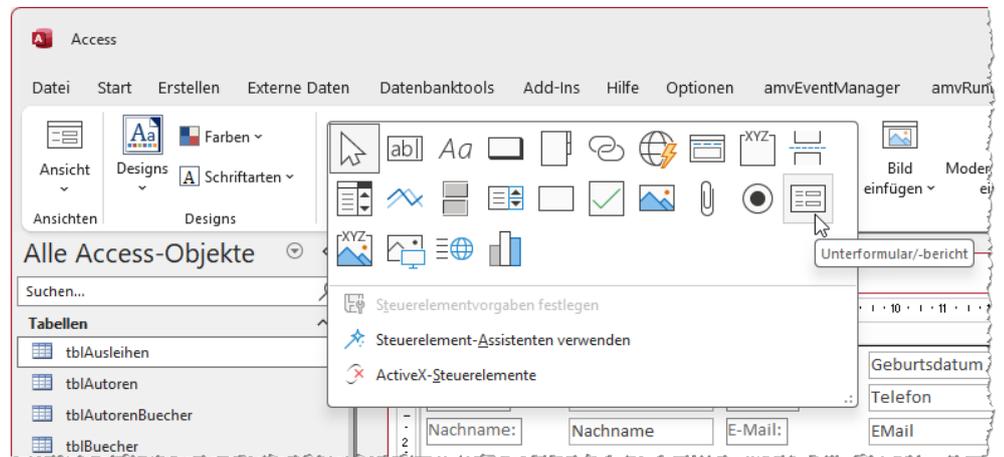


Bild 10: Auswahl des Unterformular-Steuerelements

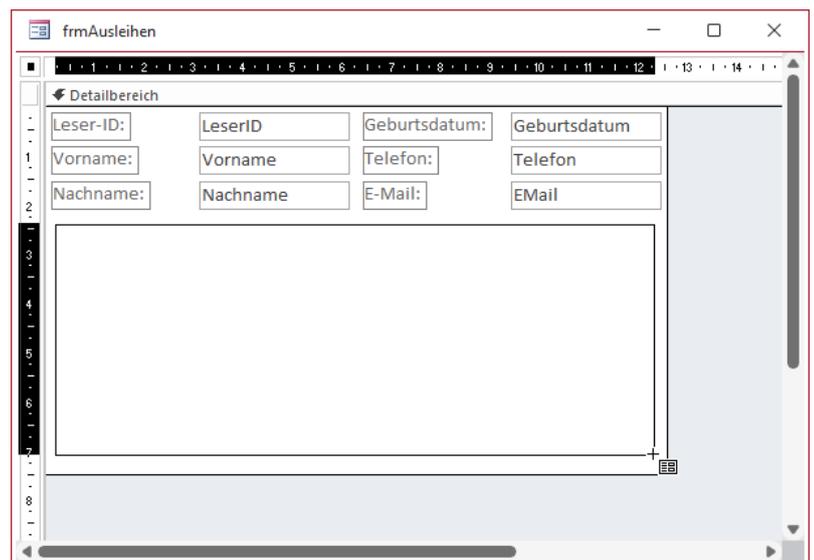


Bild 11: Aufziehen des Rahmens für das Unterformular-Steuerelement

Das Unterformular-Steuerelement wird unter einem Namen wie **Untergeordnet8** angelegt. Diesen Namen ändern wir gleich in **sfm**. Wenn wir nur ein Unterformular-Steuerelement in einem Formular verwenden, ist das völlig in Ordnung – erst wenn wir mehrere Unterformulare nutzen, sollten wir Namen verwenden, die den Inhalt repräsentieren, hier zum Beispiel **sfmAusleihen**. Wenn wir die Bezeichnung **sfm** verwenden, erhalten wir aber gleichzeitig einen anderen Namen als den des Formulars, das wir gleich in das Unterformular-Steuerelement einbetten wollen. Auch das Bezeichnungsfeld dieses Steuerelements passen wir an, beispielsweise in **Ausleihen**:

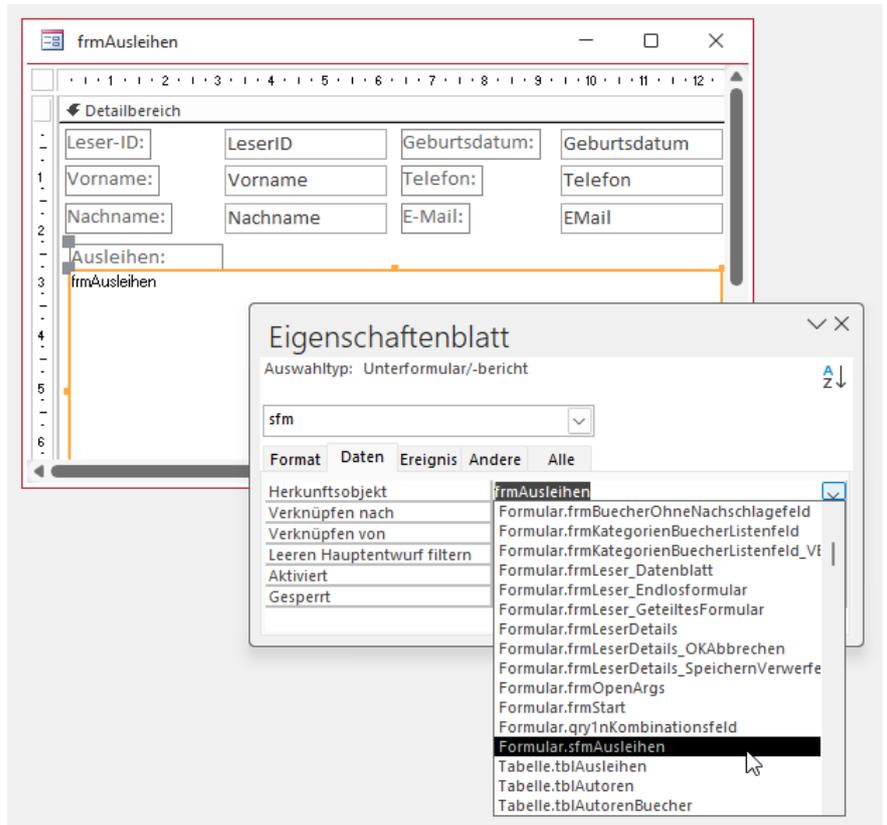


Bild 12: Einstellen des anzuzeigenden Unterformulars

Nun kommt ein wichtiger Schritt – wir weisen dem Unterformular-Steuerelement das Formular hinzu, das wir darin anzeigen wollen. Dazu stellen wir die Eigenschaft **Herkunftsobjekt** im Bereich **Daten** des Eigenschaftsblatts auf den Namen des Formulars **sfmAusleihen** ein (siehe Bild 12).

Damit erreichen wir nicht nur, dass nun das gewünschte Formular im Unterformular-Steuerelement landet, sondern es werden auch automatisch noch die beiden Eigenschaften **Verknüpfen nach** und **Verknüpfen von** auf die richtigen Werte eingestellt (siehe Bild 13). Mit diesen beiden Eigenschaften können wir einstellen, nach welchen Feldern die Daten aus dem Hauptformular und dem Unterformular synchronisiert werden sollen. Synchronisieren bedeutet in diesem Fall, dass das Unterformular nur die Datensätze anzeigen soll, die zu dem Datensatz aus dem Hauptformular gehören.



Bild 13: Die beiden Eigenschaften **Verknüpfen nach** und **Verknüpfen von** werden automatisch eingestellt

Und da wir in der Datensatzquelle des Unterformulars bereits ein Feld namens **LeserID** haben, das angibt, welchem Leser die Ausleihe zugeordnet ist, hat Access die Zuordnung genau richtig vorgenommen. Das ist allerdings kein Zufall, denn Access untersucht beim Hinzufügen eines Unterformulars beziehungs-

weise beim Auswählen des Unterformulars automatisch, ob es eine Beziehung zwischen den Datensatzquellen der beiden beteiligten Formular gibt. Und wenn es im Unterformular das Fremdschlüsselfeld einer Beziehung findet und im Hauptformular das Primärschlüsselfeld, dann stellt es die beiden Eigenschaften **Verknüpfen nach** und **Verknüpfen von** automatisch auf die Namen der beteiligten Felder ein.

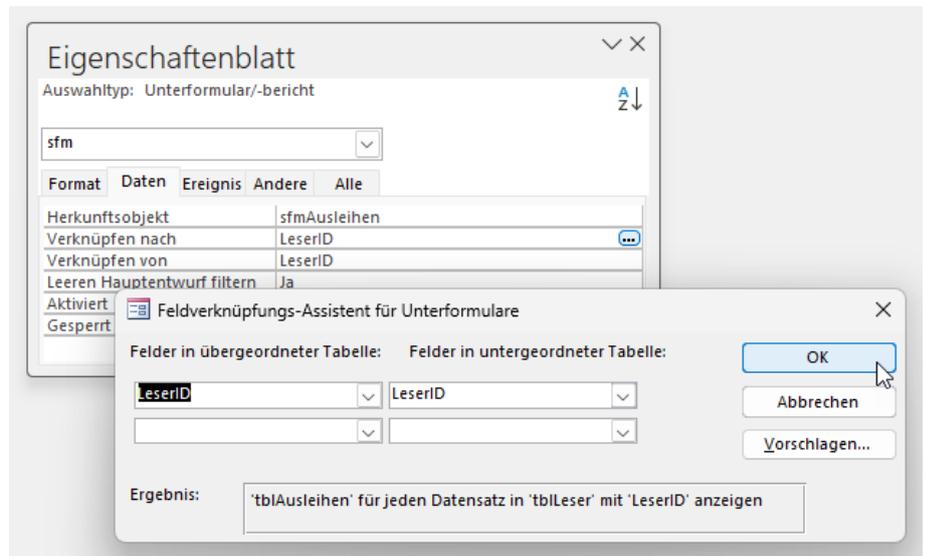


Bild 14: Einstellen der beiden Eigenschaften **Verknüpfen nach** und **Verknüpfen von** mit dem **Feldverknüpfungs-Assistent für Unterformulare**

Sollte diese Zuordnung einmal nicht klappen, hast Du zwei Möglichkeiten:

- Du weißt, welche Felder an der Beziehung beteiligt sind, und wählst diese selbstständig für die beiden Eigenschaften **Verknüpfen nach** und **Verknüpfen von** aus. Da diese in diesem Fall beide **LeserID** heißen, kannst Du nichts verkehrt machen.
- Oder Du klickst auf die Schaltfläche mit den drei Punkten und arbeitest mit dem nun erscheinenden Dialog **Feldverknüpfungs-Assistent für Unterformulare** (siehe Bild 14). Hier werden die im linken Kombinationsfeld alle Felder der Datensatzquelle des Hauptformulars angeboten und im rechten Kombinationsfeld alle Felder der Datensatzquelle des Unterformulars.

Wenn wir anschließend in die Formularansicht des Hauptformulars wechseln, erhalten wir das Formular aus Bild 15. Hier sehen wir bereits, dass nicht mehr alle Datensätze der Tabelle **tblAusleihen** angezeigt werden, sondern nur noch derjenige, der dem aktuell im Hauptformular angezeigten Leser-Datensatz zuge-

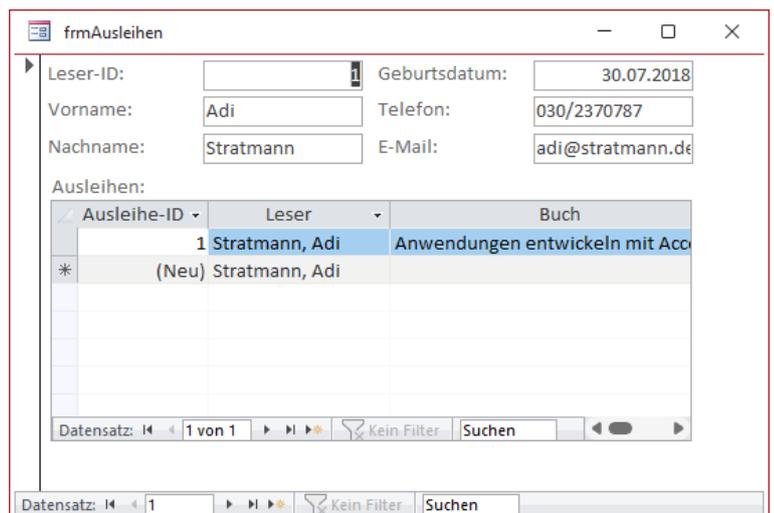


Bild 15: Hauptformular und Unterformular nach dem Wechsel in die Formularansicht

ordnet ist. Blättern wir nun durch die übrigen Datensätze im Hauptformular, sehen wir, dass sich die im Unterformular angezeigten Datensätze jeweils an den Datensatz im Hauptformular anpassen.

Damit haben wir bereits den größten Teil der Arbeit erledigt. Jetzt schauen wir uns an, was wir hier geschaffen haben.

Dabei werfen wir vor allem einen Blick auf den neuen, leeren Datensatz im Unterformular. Das Feld **Leser** zeigt hier einen Standardwert an, obwohl wir in der Tabelle selbst gar keinen Standardwert für dieses Feld hinterlegt haben. Das würde auch keinen Sinn machen, außer wir würden einen Großteil der Ausleihvorgänge immer mit der gleichen Person durchführen.

Also hat wohl Access diesen Standardwert hinzugefügt. Und das hat durchaus seinen Grund: Wenn das Unterformular nur Datensätze anzeigen soll, die mit dem aktuell im Hauptformular angezeigten Datensatz der Tabelle **tblLeser** verknüpft sind, dann ist es nur sinnvoll, ausschließlich den Wert im Feld **Leser** zur Verfügung zu stellen, der auch im Hauptformular angezeigt wird.

Wir können hier zwar auch einen anderen Leser auswählen, allerdings wird ein so ausgewählter Datensatz nur bis zur nächsten Aktualisierung unter dem aktuellen Leser angezeigt – dann erkennt Access, dass die Beziehung nicht stimmt und zeigt den Datensatz mit dem falschen Leser nicht mehr im Unterformular an.

### Primärschlüssel- und Fremdschlüsselfeld ausblenden

Jetzt, wo wir sichergestellt haben, dass das Fremdschlüsselfeld **LeserID** immer den Leser als Standardwert zugeordnet bekommt, der im Hauptformular angezeigt wird, können wir dieses Feld auch ausblenden. Schließlich sehen wir ja bereits im Hauptformular, zu welchem Leser die Ausleihe gehört.

Und wenn wir schon dabei sind, können wir auch gleich noch das Primärschlüsselfeld **AusleiheID** loswerden.

Wer gedacht hat, wir könnten diese ausblenden, indem wir ihre Eigenschaft **Sichtbar** auf den Wert **Nein** einstellen, irrt sich: Die Datenblattansicht zeigt grundsätzlich immer alle Felder an, die sich im Detailbe-

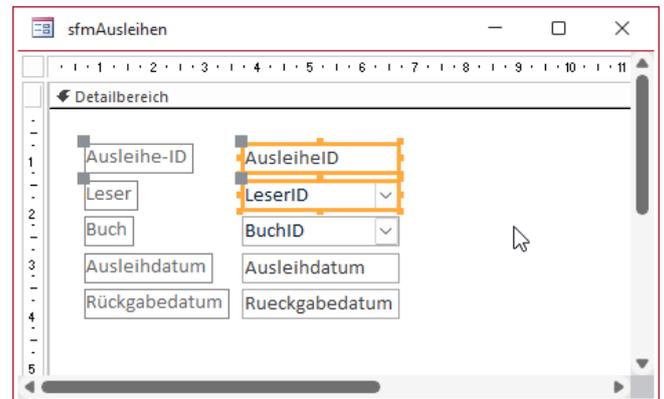


Bild 16: Entfernen der beiden Steuerelemente **AusleiheID** und **LeserID**

reich des Formularentwurfs befinden. Wir müssen die beiden Felder **AusleiheID** und **LeserID** also aus dem Entwurf entfernen (siehe Bild 16).

Gleichzeitig stellen wir die beiden Eigenschaften **Horizontaler Anker** und **Vertikaler Anker** des Unterformular-Steuer-elements auf den Wert **Beide** ein. Dadurch stellen wir sicher, dass beim Vergrößern des Hauptformulars auch das Unterformular-Steuer-element samt Inhalt vergrößert wird. So können wir wie in Bild 17 alle Felder des Unterformulars sichtbar machen.

In diesem Fall ist es bereits etwas schwieriger, das Unterformular-Steuer-element zu markieren, um die

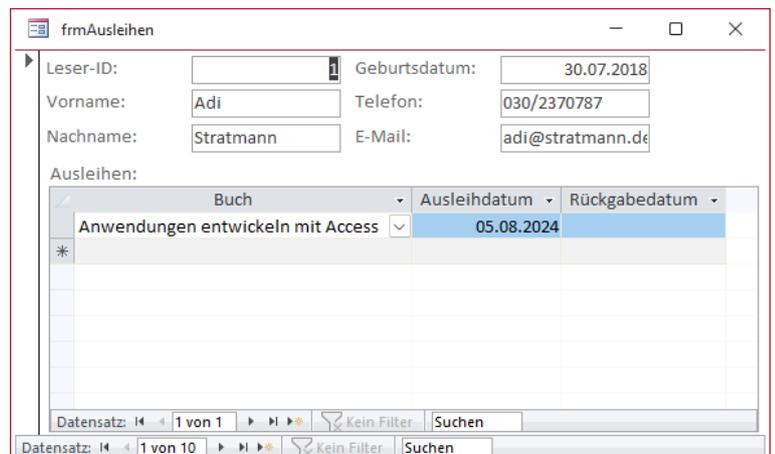


Bild 17: Unterformular mit automatischer Größenanpassung

beiden Eigenschaften einzustellen. Schnell hat man das enthaltene Unterformular statt des Unterformular-Steurelements markiert, was – um es noch einmal deutlich zu machen – zwei völlig unterschiedliche Elemente sind. Um das Unterformular-Steurelement zu markieren, hast Du verschiedene Möglichkeiten:

- Mausklick auf irgendeinen anderen Bereich im Hauptformular und anschließend auf das Unterformular,
- Aufziehen eines Rahmens, der außerhalb des Unterformular-Steurelements beginnt und im Unterformular-Steurelement endet oder
- Auswahl des Unterformular-Steurelements im Eigenschaftensblatt.

Letztlich sehen wir das mit einer Markierung versehene Unterformular wie in und können dann die beiden Eigenschaften **Horizontaler Anker** und **Vertikaler Anker** wie gewünscht einstellen (siehe Bild 18).

### Unterformular aus dem Navigationsbereich hineinziehen

Nun löschen wir das Unterformularsteurelement aus dem Hauptformular und schauen uns den zweiten, aus unserer Sicht effizienteren Weg zum Einfügen eines Unterformulars in ein Hauptformular an.

Dazu suchen wir im Navigationsbereich das als Unterformular einzusetzende Formular **sfmAusleihen** und ziehen es in den freien Bereich des Hauptformulars (siehe Bild 19).

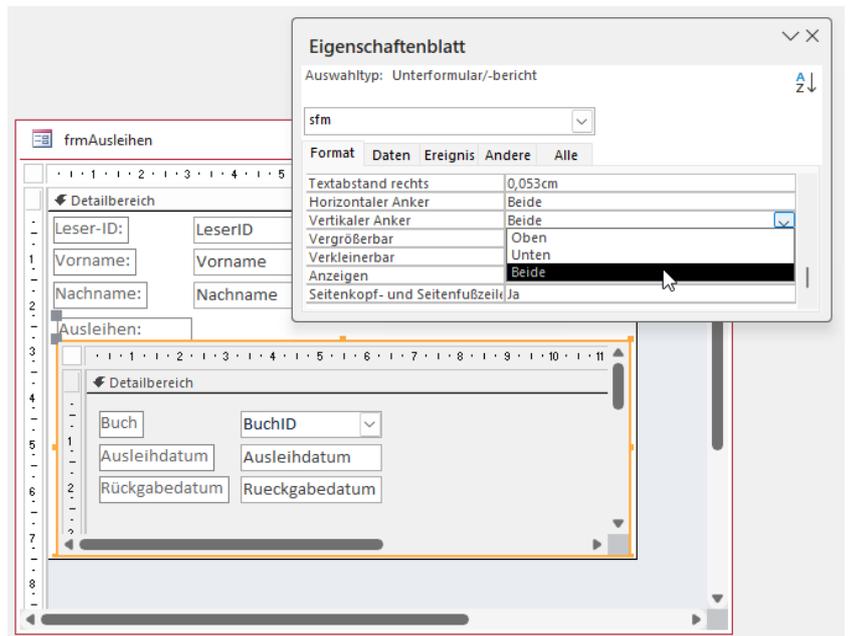


Bild 18: Einstellen der Größenanpassung des Unterformulars

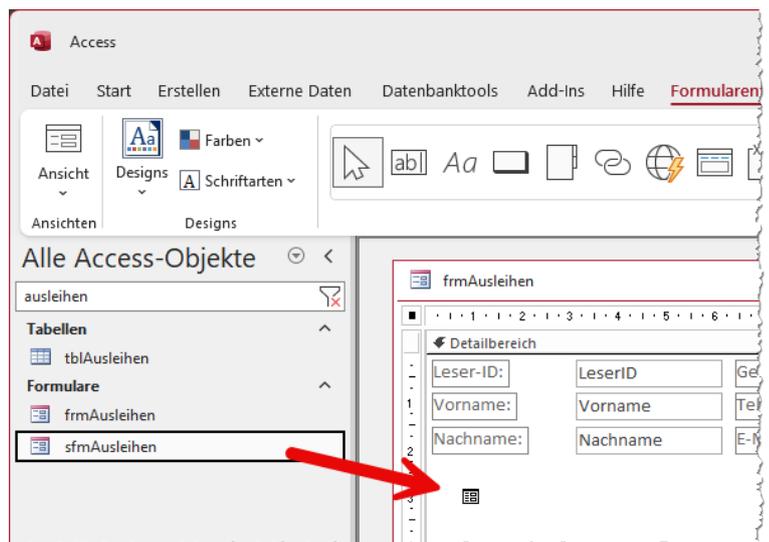


Bild 19: Hinzufügen eines Unterformulars aus dem Navigationsbereich

Lassen wir es fallen, erhalten wir in einem Schritt ein Unterformularsteurelement, das direkt mit dem richtigen Unterformular gefüllt ist. Hier gilt es nun zunächst zu beachten, dass das Unterformularsteurelement genau den gleichen Namen erhält wie das im Unterformularsteurelement enthaltene Unterformular, hier also **sfmAusleihen**. Das führt in vielen Fällen, gerade bei Anfängern, zu Missverständnissen. Sie

glauben dann, wenn sie das Element `Me!sfmAusleihen` referenzieren, dass sie direkt auf das im Unterformular angezeigte Unterformular zugreifen würden. Dann sind sie verwundert, dass beispielsweise `Me!sfmAusleihen!BuchID` einen Fehler liefert.

Tatsächlich benötigen wir jedoch den folgenden Ausdruck, um beispielsweise einen der Feldinhalte aus dem Unterformular in einem weiteren Textfeld auszugeben:

```
Forms!frmAusleihen!sfmAusleihen.Form!BuchID
```

Deshalb kann es sinnvoll sein, das Unterformularsteuerelement auch in diesem Fall nachträglich umbenennen, beispielsweise einfach in `sfm`. Das sollte man allerdings nur machen, wenn das Formular nur ein Unterformular enthält – anderenfalls können wir ja nicht mehrere Unterformularsteuerelemente `sfm` nennen.

## Feldwerte beim Anlegen von Datensätzen im Unterformular automatisch setzen

Bei dem absoluten Standardbeispiel für m:n-Beziehungen in Haupt- und Unterformular, den Rechnungen und Rechnungspositionen, führt man meist noch eine Aktion durch, wenn eine neue Rechnungsposition angelegt und dabei der Artikel dieser Position ausgewählt wird. Hier trägt man nämlich beispielsweise den Einzelpreis und den Mehrwertsteuersatz des Artikels ein, damit der entsprechende Wert in der Artikeltable sich gegebenenfalls im Laufe der Zeit ändern kann. Die Rechnungsdaten sollen jedoch erhalten bleiben, deshalb erfassen wir diese Daten in eigenen Feldern in der m:n-Verknüpfungstabelle.

Auch in unserem Beispiel haben wir ein Feld, das wir gleich beim Anlegen eines neuen Datensatzes im

Unterformular füllen können – nämlich das **Ausleihdatum**. Dieses wollen wir, ausgehend davon, dass die Ausleihe an dem Tag eingetragen wird, an dem sie auch stattfindet, auf den aktuellen Tag einstellen.

Dazu fügen wir dem Unterformular eine VBA-Prozedur hinzu, die durch das Ereignis **Nach Aktualisierung** des Feldes `BuchID` ausgelöst wird. Dieses Ereignis feuert immer, wenn der Benutzer ein neues Buch in einem der Datensätze im Unterformular ausgewählt hat.

Wir legen es an, indem wir das Hauptformular mit dem Unterformular in die Entwurfsansicht versetzen. Dann klicken wir auf das Feld `BuchID` im Unterformular und zeigen im Eigenschaftenblatt die Seite **Ereignisse** an. Hier wählen wir für die Ereigniseigenschaft **Nach Aktualisierung** den Wert **[Ereignisprozedur]** aus und klicken rechts auf die Schaltfläche mit den drei Punkten (siehe Bild 20).

Dies öffnet den VBA-Editor und zeigt direkt die frisch angelegte Ereignisprozedur `BuchID_AfterUpdate` an (siehe Bild 21).

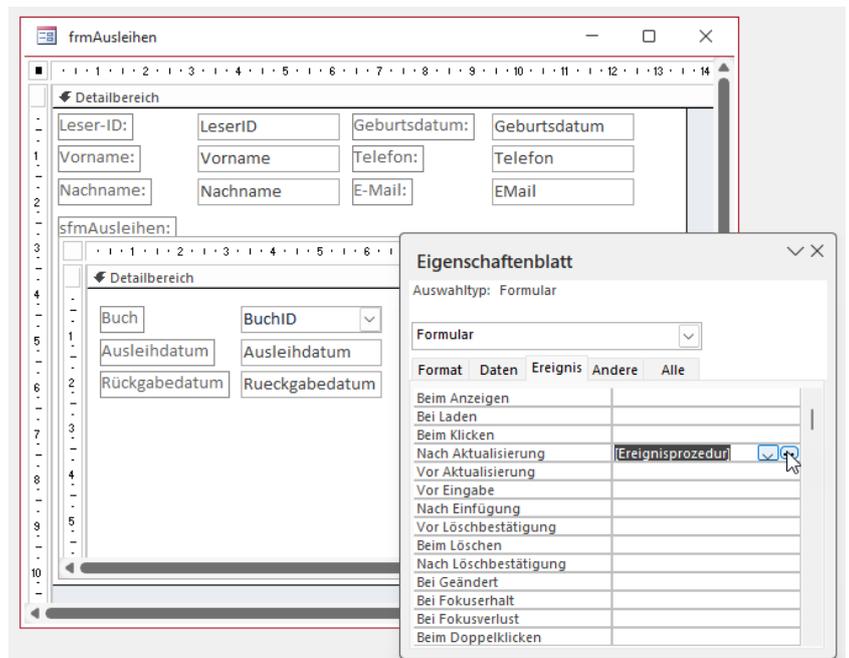


Bild 20: Hinzufügen eines Ereignisses zum Feld `BuchID`

Diese füllen wir mit der folgenden Anweisung:

```
Private Sub BuchID_AfterUpdate()  
    Me!Ausleihdatum = Date  
End Sub
```

Diese weist einfach dem Feld Ausleihdatum das Ergebnis der Funktion `Date` zu, die das aktuelle Datum ermittelt.

Wechseln wir nun zurück in die Formularansicht und wählen in einem neuen Datensatz ein neues Buch aus, legt die Prozedur das aktuelle Datum im Feld `Ausleihdatum` an (siehe Bild 22).

Im Falle des Beispiels mit den Rechnungen ist die Vorgehensweise etwas komplizierter, aber auch überschaubar: Hier müssten wir mit der `DLookup`-Anweisung erst noch herausfinden, welcher Einzelpreis und welcher Mehrwertsteuersatz in der Artikeltabelle für den der Bestellposition hinzugefügten Artikel gelten. Letztlich benötigen wir dazu aber auch nur einen einzigen Funktionsaufruf je zu füllendem Feld.

## Zusammenfassung und Ausblick

Mit der hier vorgestellten Vorgehensweise haben wir ein praktisches Formular mit einem Unterformular erstellt, in dem wir die Daten zweier per m:n-Beziehung verknüpfter Tabellen verwalten können. Außerdem haben wir gleich noch gezeigt, wie man automatisch die Daten eines neuen Datensatzes anpassen kann.

Beim Einfügen des Unterformular haben wir zwei Methoden vorgestellt: Bei der ersten haben wir zunächst ein Unterformularsteuerelement zum Formular hinzugefügt und dann das anzuzeigende Unterformular über die Eigenschaft `Herkunftsobjekt` ausgewählt. Bei der zweiten Methode haben wir einfach das Unterfor-

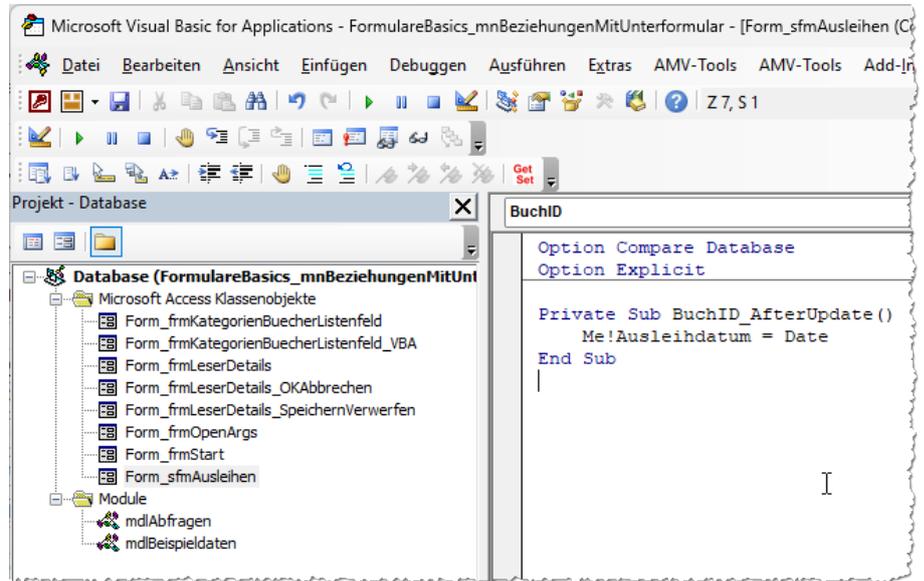


Bild 21: Füllen des Ereignisses für das Feld `BuchID`

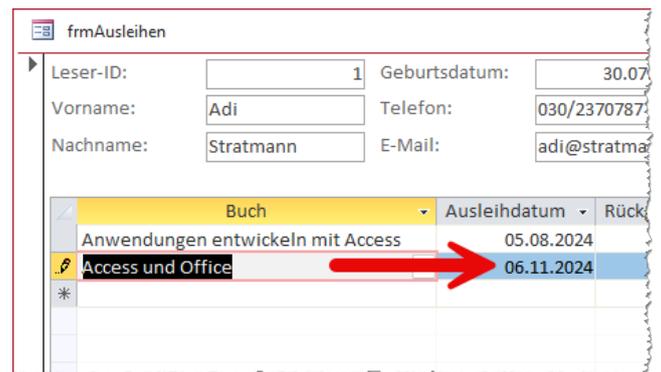


Bild 22: Das Ausleihdatum wird automatisch gefüllt.

mular aus dem Navigationsbereich in das Hauptformular gezogen und ein ähnliches Ergebnis erhalten.

Beim Eingeben von Daten in eine Kombination aus Haupt- und Unterformular kann es jedoch leicht zu Problemen kommen. Wenn wir nämlich Daten im Unterformular eingeben, ohne dass wir einen Datensatz im Hauptformular angelegt haben, verschwinden die Datensätze aus dem Unterformular scheinbar im Nirvana. Wie wir diese umschiffen, zeigen wir in einem weiteren Artikel namens \*\*\*\*\* (\*\*\*\*\*),