

In diesem Formular können keine Datenfelder bearbeitet werden. Darum gibt es auch keine Schaltflächen zum Anlegen, Speichern und Löschen von Datensätzen. Der einzige Zweck dieses Formulars besteht darin, einen Auftrag anhand des Kundennamens zu finden.

Bitte beachten Sie die Spalte *Status* im Unterformular. Auch hier handelt es sich wieder um eine Einzelzuordnung mithilfe eines Kombinationsfeldes; nur dass sich das Kombinationsfeld jetzt im Unterformular befindet. Das ändert aber nichts an der Vorgehensweise zu seiner Erstellung.

Jetzt können Sie auch noch einmal zu Abbildung 3.15 zurückblättern. Ich hatte dort auf das Kombinationsfeld zur Auswahl der Kontaktart hingewiesen, ohne zu erläutern, wie es erstellt wird. Nach der Lektüre dieses Abschnitts über die Einzelzuordnung mit Kombinationsfeldern ist das bestimmt etwas klarer geworden!

Sie haben jetzt also gesehen, wie man ein Formular erstellt, das

- Daten aus einer einzigen Tabelle enthält (ein Stammdatenformular),
- aus der »1er-Richtung« auf eine 1:n-Beziehung blickt (ein Kunde mit allen seinen Aufträgen),
- aus der »n-Richtung« auf eine 1:n-Beziehung blickt (ein Auftrag mit dem Kunden, der ihn erteilt hat).

Was noch fehlt, ist ein Formular, mit dem man aus einer Liste von Objekten **mehrere** auswählen und einem anderen Objekt zuordnen kann – z. B. aus der Liste aller Vereinsmitglieder diejenigen auswählen, die an einem Training teilnehmen. Ich erläutere das in Kapitel 9 im Abschnitt »Eine Tabelle mit einer 1:n- und einer n:1-Beziehung«.

Die beiden Grundbausteine

Bevor wir weiter machen, möchte ich noch eine ganz, ganz wichtige Zwischenzusammenfassung geben. In den letzten beiden Abschnitten haben wir zwei Formulartypen entwickelt, die von grundsätzlicher Bedeutung sind.

- Im Abschnitt »Ein Formular mit Unterformular« gehörten zu einem Objekt (Kunde) MEHRERE andere Objekte (Kontakte). Infolge dessen enthielt das Formular ein Unterformular zur Anzeige der anderen Objekte (Kontakte).
- Im Abschnitt »Einzelzuordnung« gehörte zu einem Objekt (Auftrag) EIN anderes Objekte (Kunde). Infolge dessen enthielt das Formular ein Kombinationsfeld zur Anzeige und Auswahl des anderen Objektes (Kunde).

Die Abbildung 3.20 zeigt das noch einmal am Beispiel der Relation zwischen Kunde und Auftrag.

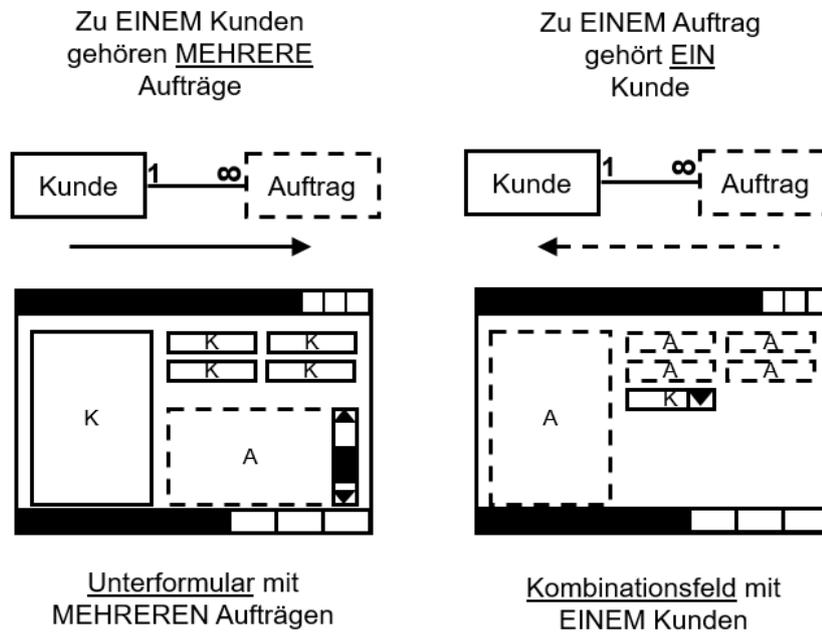


Abbildung 3.20: Zu jeder 1:n-Beziehung gibt es zwei unterschiedliche Blickrichtungen und damit zwei unterschiedliche Formularlayouts (K = Kunde, A = Auftrag)

Es gibt also grundsätzlich zwei unterschiedliche Blickrichtungen in einer 1:n-Beziehung – von der »1-Seite« zu »n-Seite« bzw. von der »n-Seite« zu »1-Seite« (fast wie bei den Menschen ;-). Resultierend daraus enthält das Formularlayout entweder ein Unterformular zur Anzeige mehrerer Objekte oder ein Kombinationsfeld zur Anzeige und zur Auswahl eines Objektes.

Wenn Sie jetzt aber einen Blick auf die Abbildung 3.3 werfen, dann sehen Sie, dass es außer 1:n-Beziehungen gar nichts anderes gibt. Die m:n-Beziehungen aus dem logischen Datenmodell werden ja im physischen Datenmodell in zwei 1:n-Beziehungen aufgelöst und 1:1-Beziehungen sind zwar prinzipiell möglich, aber meistens falsch modelliert oder nutzlos (siehe »Modellbesonderheiten und -erweiterungen« in Kapitel 2).

Das bedeutet also, dass die Abbildung 3.20 zwei Grundbausteine von Formularen zeigt, aus denen (fast) alle denkbaren Formulare aufgebaut sind. Sehen Sie sich zum Beispiel einmal die Abbildung 3.19 an: Zu einem Kunden gehören mehrere Aufträge. Also enthält das Formular ein Unterformular für die Aufträge. Nun gehört aber zu einem Auftrag (im Unterformular!) wiederum ein Status. Also gibt es im Unterformular in jeder Zeile mit jeweils einem Auftrag ein Kombinationsfeld zur Auswahl des Status. Hier wurden also die beiden Grundbausteine miteinander kombiniert!

Blättern Sie dazu auch mal das Kapitel 9 durch. Dort erläutere ich den Aufbau von Formularen für verschiedene Konfigurationen von 1:n-Beziehungen. In den Layouts werden Sie die beiden Grundbausteine aus Abbildung 3.20 immer wieder entdecken.

Wenn Ihnen dieses Prinzip einmal klar geworden ist, wird es Ihnen viel leichter fallen, Formulare zu entwickeln!