

"Welche Tabellen brauche ich?"

"Habe ich in meiner Datenbank die richtigen Tabellen?"

Die Antwort auf diese Fragen liefert ein Datenmodell.

Dieser Artikel zeigt, wie sich ein gutes Datenmodell entwickeln lässt.

DENN: Erst ein gutes Datenmodell erlaubt es, Access sinnvoll zu nutzen.

Ein Datenmodell ist normalerweise

(1) eine Grafik

(2) eine Sache, für die man 30 Buchseiten zur Erläuterung braucht.

Hier kommt eine praktische Anleitung ohne Grafik und auf ca. 3 Seiten!

Also - nehmen wir mal an, in einer Bibliothek werden Bücher an Leser verliehen. Die Bücher sind von Verlagen und stehen in Regalen.

1. Schritt: Formulieren Sie das Problem in Sätzen der Form Substantiv - Verb - Substantiv

- Leser leihen Bücher.
- Bücher sind von Verlagen.
- Bücher stehen in Regalen.

REGEL: Substantive werden zu Tabellen; Verben werden zu Beziehungen - d.h.:

Tabellen: tblLeser, tblBuch, tblVerlag, tblRegal

Beziehungen: tblLeser - tblBuch, tblBuch - tblVerlag, tblBuch - tblRegal

ACHTUNG: Alle Bezeichnungen für Tabellen und deren Spalten ohne Umlaute (ä, ü, ö), ohne Leerzeichen und ohne ß schreiben!

2. Schritt: Formulieren Sie für jede Beziehung zwei Sätze

- EIN Leser leiht MEHRERE Bücher.
- EIN Buch wird von MEHREREN Lesern ausgeliehen.

- EIN Buch steht in EINEM Regal.
- In EINEM Regal stehen MEHRERE Bücher.

- EIN Buch ist von EINEM Verlag.
- Von EINEM Verlag gibt es MEHRERE Bücher.

Damit haben wir folgende Beziehungen: (" $\leftarrow\rightarrow$ " = "1:n", " $\leftarrow\leftarrow\rightarrow$ " = "m:n")

- tblRegal $\leftarrow\rightarrow$ tblBuch
- tblVerlag $\leftarrow\rightarrow$ tblBuch
- tblLeser $\leftarrow\leftarrow\rightarrow$ tblBuch

Vielleicht haben Sie sich gewundert, warum EIN Buch an MEHRERE Leser ausgeliehen werden kann. EIN Buch kann doch immer nur bei EINEM Leser sein!? Das ist nach gesundem Menschenverstand auch richtig - datenbanktechnisch aber nicht, wenn nämlich die Zeit ins Spiel kommt: Das Buch kann heute bei Meier sein; vorige Woche hatte es Lehmann und für nächste Woche hat es Krause vorbestellt.

Weitere Beispiele aus anderen Anwendungsgebieten:

- EIN Schiff ist in EINEM Hafen.
- In EINEM Hafen liegen MEHRERE Schiffe.

Stimmt? Scheinbar ja - aber das Schiff kann ja gestern in Hamburg gewesen sein, ist heute in Antwerpen und morgen in London.

- EIN Leihwagen ist bei EINEM Kunden.
- EIN Kunde leiht MEHRERE Wagen.

Stimmt? Scheinbar ja - aber der Leihwagen kann ja gestern bei Schmidt gewesen sein, ist heute bei Hauptmann und morgen bei Schubert.

ALSO: Immer wenn es um das Aufheben von Vergangenheitsdaten oder um die Planung für die Zukunft geht, ist eine scheinbare 1:n-Beziehung in Wirklichkeit eine m:n-Beziehung! Die Zeitangaben gehören dann zusammen mit den beiden Fremdschlüsseln in die Zwischentabelle.

3. Schritt: Anlegen der Fremdschlüssel

Dazu zwei **REGELN:**

- (1) Bei einer 1:n-Beziehung wandert der Primärschlüssel der "1"-Seite als Fremdschlüssel auf die "n"-Seite.
- (2) Bei einer m:n-Beziehung wandern beide Primärschlüssel als Fremdschlüssel in eine neue, dritte Tabelle zwischen den beiden Tabellen.

Also:

- regal_id als Fremdschlüssel regal_id_f in Tabelle tblBuch
- verlag_id als Fremdschlüssel verlag_id_f in Tabelle tblBuch
- leser_id als Fremdschlüssel leser_id_f in Tabelle tblLeser_Buch
- buch_id als Fremdschlüssel buch_id_f in Tabelle tblLeser_Buch

Es stellt sich also heraus, dass Sie neben den offensichtlichen Tabellen tblLeser, tblBuch, tblRegal und tblVerlag wegen der m:n-Beziehung zwischen Leser und Buch noch eine weitere Tabelle brauchen: tblLeser_Buch. Ich nenne diese Tabellen "Zwischentabellen", weil sie zwischen zwei Tabellen eingeschoben werden.

4. Schritt: Überlegen Sie sich, ob Sie bestimmte Mengen oder Zeiten speichern wollen - schauen Sie sich dazu Ihre 3-Wort-Sätze noch einmal an:

- Leser leihen Bücher (Zeit: "von wann bis wann?")
- Bücher sind von Verlagen.
- Bücher stehen in Regalen.

Weitere Beispiele

- Aufträge enthalten Artikel (Menge: "Wie viele?"))
- Patienten erhalten Medikamente (Menge: "Wie viele?", Zeit: "Wann?")
- Fahrer fahren LKWs (Zeit: "Wann?")

Diese Mengen und Zeiten gehören in die Zwischentabellen! Damit hat die Zwischentabelle tblLeser_Buch dann neben den beiden Fremdschlüsselspalten leser_id_f und buch_id_f noch die Spalten leserbuch_von und leserbuch_bis.

Fertig! Jetzt können Sie Access starten, die Tabellen mit den dazugehörigen Spalten anlegen (Primärschlüssel als Autowert, Fremdschlüssel als Zahl / LongInteger) und schließlich per drag&drop die Beziehungen definieren ("Mit referentieller Integrität!"). Dann haben Sie ein sauberes Datenmodell und bedeutend weniger Probleme beim Erstellen der Abfragen, Formulare und Berichte! **Garantiert!**

ERGÄNZUNGEN:

Besonderheit: 1:1-Beziehungen

Es kann sein, dass Sie bei der Formulierung Ihrer Sätze auf so etwas stoßen:

- Zu EINEM Auftrag gehört EINE Rechnung.
- Zu EINER Rechnung gehört EIN Auftrag.

Das würde bedeuten, dass zwischen Auftrag und Rechnung eine 1:1-Beziehung besteht. Überlegen Sie nochmal, ob die Sätze wirklich stimmen! Solche Beziehungen sind in den allermeisten Fällen nicht sinnvoll!

Erstellen Sie vielleicht Teilrechnungen:

- Zu EINEM Auftrag gehören MEHRERE (Teil-)Rechnungen.

oder Sammelrechnungen:

- Zu EINER (Sammel-)Rechnung gehören MEHRERE Aufträge.

Wenn nicht, dann brauchen Sie nicht zwei Tabellen tblAuftrag und tblRechnung, sondern nur eine Tabelle tblAuftrag. Sie enthält dann neben Spalten wie auf_nummer und auf_datum noch Spalten wie auf_rechnummer und auf_rechdatum!

Besonderheit: Dreifach-Beziehungen

Es kann sein, dass Sie bei der Formulierung Ihrer Sätze auf so etwas stoßen:

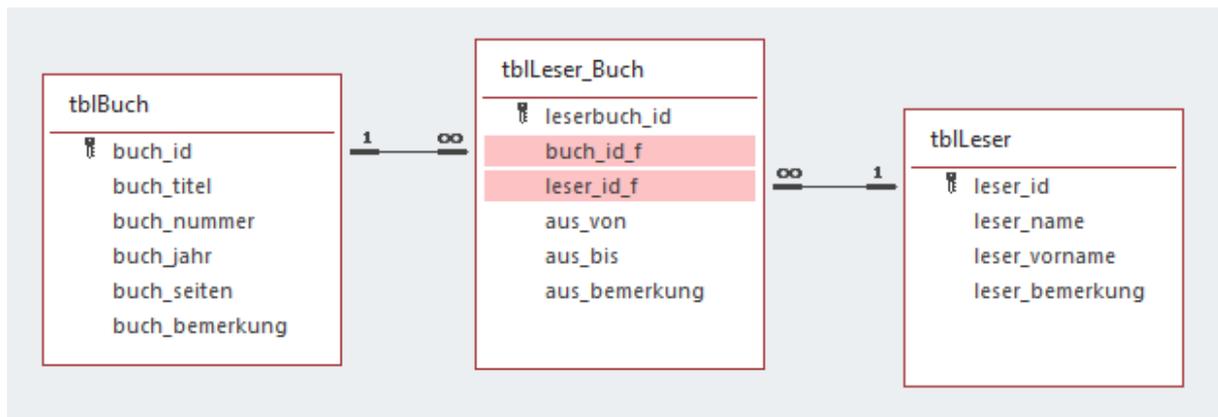
- EIN Leser erhält für die Ausleihe EINES Buches EINE Mahnung.

Dieser Satz enthält die Namen von drei Objekten und Sie können keines davon weglassen, ohne dass der Satz seinen Sinn verliert! Solche Sätze sind nicht leicht zu finden - aber sie sind u.U. ganz wichtig für Ihr Datenmodell. Sie sollten daher Ihre "Zwei-Objekt-Sätze" immer nochmal unter dem Aspekt überprüfen: Gehört da evtl. noch ein drittes Objekt dazu, damit der Sachverhalt wirklich realistisch erfasst wird? "EIN Leser erhält EINE Mahnung." Ist das wirklich so? Nein! Er erhält eine Mahnung wegen der Ausleihe eines Buches!

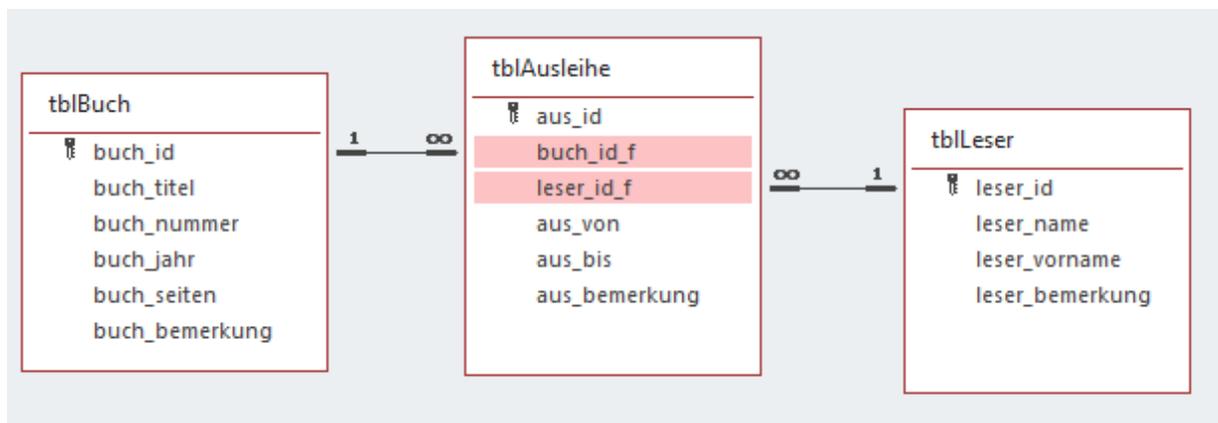
Bei einer Dreifach-Beziehung (z.B. Leser - Buch - Mahnung; s.o.!) bilden zunächst einmal zwei der drei Objekte eine "normale" m:n-Beziehung - also z.B.:

- EIN Leser leiht MEHRERE Bücher aus.
- EIN Buch wird von MEHREREN Lesern ausgeliehen.

Das ergibt also zunächst eine "normale" m:n- Beziehung zwischen Leser und Buch; folglich brauchen wir eine Zwischentabelle tblLeser_Buch.



Wie hängt jetzt die Mahnung damit zusammen? Um das herauszufinden, sollten Sie zunächst versuchen, den sperrigen Namen der Zwischentabelle "tblLeser_Buch" durch einen anderen, anschaulicheren Begriff zu ersetzen, sonst lassen sich damit schlecht Sätze bilden. In unserem Bibliotheksbeispiel bietet sich für "tblLeser_Buch" der Begriff "tblAusleihe" an!



Test:

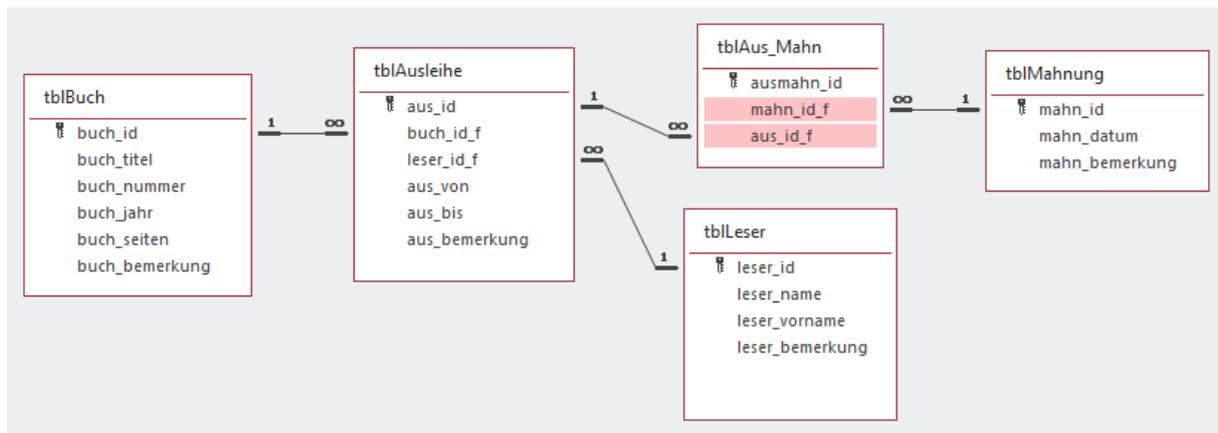
- Zu EINEM Leser gehören MEHRERE Ausleihen.
- ZU EINER Ausleihe gehört EIN Leser.
- Zu EINEM Buch gehören mehrere Ausleihen.
- Zu EINER Ausleihe gehört EIN Buch.

Stimmt! Nun können wir über die Mahnung nachdenken. Wie hängt sie mit der Ausleihe zusammen?

- Zu EINER Ausleihe gibt es MEHRERE Mahnungen (erste, zweite, dritte Mahnung)
- Zu EINER Mahnung gehören MEHRERE Ausleihen.(denn beachte: Zu EINER Ausleihe gehört nur EIN Buch! Und wir können ja mehrere Bücher mit einer Mahnung mahnen!)

Also gibt es eine m:n-Beziehung zwischen Ausleihe und Mahnung! Also brauchen wir eine weitere Zwischentabelle "tblAus_Mahn"! Genau genommen ist das jetzt eine Zwischentabelle

zwischen einer Tabelle (tblMahnung) und einer (ehemaligen) Zwischentabelle (tblAusleihe_Buch).

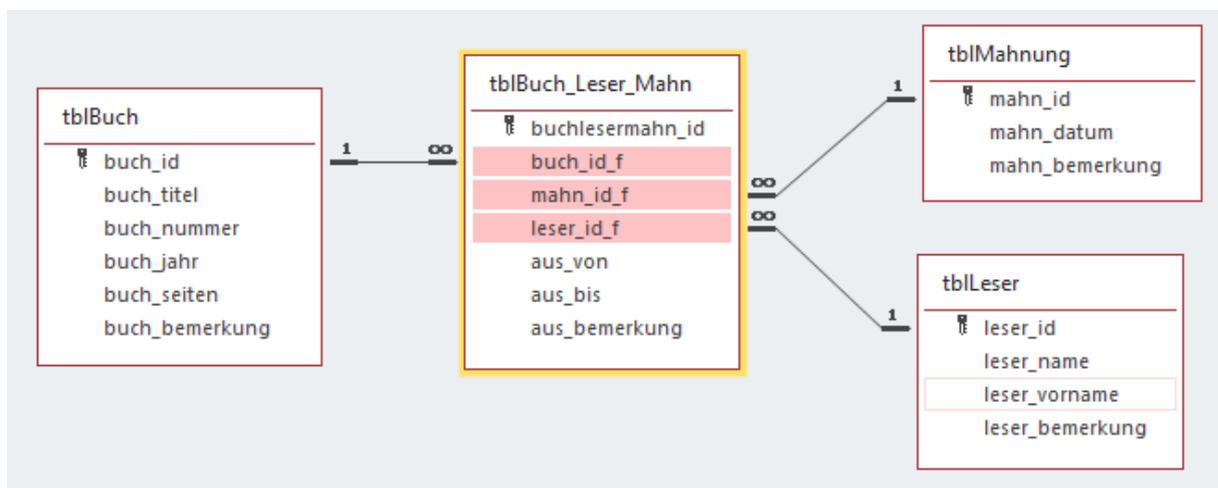


Das Beispiel mit der Mahnung war besonders schwierig, weil die dritte Beziehung (von Ausleihe zu Mahnung) eine m:n-Beziehung war. Nehmen wir mal unrealistischerweise an, es würde zu jeder Ausleihe immer nur eine einzige Mahnung geschrieben. Wie gesagt - das ist unrealistisch. Aber ich will Sie nicht mit noch einem weiteren Beispiel verwirren. Es geht mir nur darum, das Prinzip darzustellen.

Also:

- Zu EINER Ausleihe gehört EINE Mahnung.
- Zu EINER Mahnung gehören MEHRERE Ausleihen.

Dann sieht das Datenmodell so aus, wie in dem Bild unten. Dort habe ich der Zwischentabelle wieder den "unhandlichen" Namen tblBuch_Leser_Mahn gegeben, um ihren Charakter als Zwischentabelle besonders zu betonen. Sie ist jetzt eine Zwischentabelle zwischen DREI Tabellen: tblBuch, tblLeser und tblMahnung. Folgerichtig enthält sie jetzt DREI Fremdschlüssel: buch_id_f, leser_id_f und mahn_id_f. Das ist völlig analog zu der "normalen" Zwischentabelle zwischen zwei anderen Tabellen, wie wir sie bereits kennen!



Mittlerweile wird das also richtig kompliziert.

Es bleibt aber dabei: Sie müssen kurze, knappe Sätze nach dem Schema "Zu EINEM x gehört EIN bzw. gehören MEHRERE y" bilden und diese Sätze in Beziehungen zwischen Tabellen umsetzen.

Die schlechte Nachricht ist: Das ist nicht einfach!

Die gute Nachricht ist: Die von mir beschriebene Vorgehensweise führt Sie (ziemlich) sicher zum Ziel – zu einem richtig guten Datenmodell!!

PS: Man kann ja auch viel aus Beispielen lernen. Darum sehen Sie sich mal diese Webseite an: <https://datamodels.databases.biz>