

# Die Schwierigkeiten mit der 1-zu-n-Beziehung

---

*Datenmodellierung*

*Thilo Riegel, München*

## Allgemeines zu den Beziehungen

Die große Stärke relationaler Datenbanken liegt, wie der Name schon sagt, in der Tatsache, dass man nicht nur Entitäten, sondern auch die Relationen zwischen ihnen verwalten kann, und das auf maximal flexible Art.

So gibt es folgende drei Grundarten von Relationen:

- 1-zu-1-Beziehung: Jedes Element in Tabelle A steht mit 0 oder genau einem Element in Tabelle B in Beziehung. Strenge Version: genau ein Element, 0 ist verboten.
- 1-zu-n-Beziehung: Jedes Element in Tabelle A steht mit beliebige vielen Elementen in Tabelle B in Beziehung (0 auch erlaubt). Strenge Version: 0 verboten.
- m-zu-n-Beziehung: Wie 1-zu-n, jedoch gleichzeitig auch umgekehrt.

In der Praxis des Datenbank-Designs zeigt sich, dass 1-zu-1-Beziehungen relativ selten sind, und 1-zu-n-Beziehungen sind relativ häufig anzutreffen. Allerdings sieht man bei genauerem Hinsehen, dass die angeblichen 1-zu-n-Beziehungen eigentlich gar keine sind, oder genauer gesagt: zwar als solche definiert wurden, aber zweckmäßigerweise hätte man lieber m-zu-n gewählt.

## Vorsicht vor den 1-zu-n-Beziehungen!

Grundregel: 1-zu-n-Beziehungen sind gefährlich und generell beim Datenmodell-Design zu vermeiden. 1-zu-n-Beziehungen gibt es in „freier Wildbahn“ nur sehr selten. In fast allen praktisch relevanten Fällen ist eine m-zu-n-Verbindung vorzuziehen.

Warum? Das Design eines Datenmodells ist ein sehr wichtiger und sehr grundlegender Schritt im Systemdesign. Und es ist eine Sache, bei der man lieber nicht schlampen sollte, denn jeder Fehler, den man hier macht, wird nachher garantiert sehr teuer. Datenmodelle sollen die *fachlichen Gegebenheiten* angemessen nachbilden, also mit anderen Worten: die „Wirklichkeit“. „Kein Problem“, denkt man, „das kennen wir ja schon sehr gut aus der betrieblichen Praxis“. Allzu oft zeigt sich jedoch, dass das eben nicht der Fall ist, und dass die Verhältnisse eben nur ungefähr bekannt sind, aber nicht genau. Oder auch: es ist bekannt für alle „wichtigen“ Fälle, also für 95%. Der Rest sind Ausnahmen. Und darauf kommt es beim Datenmodell an: es muss *genau* sein, und es muss *alle* Fälle abdecken. 95% oder 98% genügen nicht. 100% sind die Norm. Punkt.

Ja, ich weiß schon, Sie werden jetzt einwenden, dass man sich in der betrieblichen Praxis diesen Luxus nicht leisten könne, und es sei doch allgemein bekannt, dass man mit dem Pareto-Prinzip (auch 80-20-Regel genannt) erfahrungsgemäß ganz gut fahre. Das stimmt, da widerspreche ich nicht. Aber

ich möchte einschränken: mit dem Pareto-Prinzip fährt man *im Allgemeinen* ganz gut, und es ist in der Tat so, dass man sich *in den meisten Fällen* den Luxus einer 100%-Lösung nicht leisten kann auch nicht zu leisten braucht. Aber eben nicht in allen Fällen. Es gibt Sachen, bei denen sind 80% oder auch 95% einfach nicht ausreichend. Das sind nicht viele Sachen, zugegeben, aber es gibt sie eben schon. Bei diesen Sachen sind nur 100% akzeptabel. Ansonsten lässt man es lieber gleich bleiben.

Es gibt das schöne (und etwas abgelutschte) Beispiel von der Hebamme: wie oft darf eine Hebamme im Laufe eines Berufslebens ein ihr anvertrautes Baby fallen lassen? Welche Quote gilt da noch als akzeptabel? Eben. Oder auch das Beispiel der amerikanischen Fallschirmspringer im zweiten Weltkrieg:<sup>1</sup> Bei der Air Force waren bestimmte Leute des Bodenpersonals dafür zuständig, die Fallschirme für Ihre Kameraden zu falten. Leider war die Rate der Fallschirme, die sich nicht öffneten, inakzeptabel hoch. Der zuständige Kommandant befahl darauf hin, dass jeder Fallschirm-Falter in bestimmten Abständen einen seiner selbst gefalteten Fallschirme testen müsse. Die Rate ging darauf hin dramatisch zurück: so gut wie alle Fallschirme öffneten sich!

Betrachten Sie Ihr Datenmodell als Ihren Fallschirm. Er muss sich öffnen. Immer. Ohne Ausnahme. Denn es hängt im Unternehmensumfeld einfach zu viel dran.

Das bringt uns zurück zur eigentlichen Frage: warum sind 1-zu-n-Beziehungen gefährlich und mit-hin zu vermeiden? Weil eine technische 1-zu-n-Beziehung in der Datenbank immer eine *strenge* 1-zu-n-Beziehung ist. Sie kennt keine Ausnahmen. Sie ist „gnadenlos“. Das ist ungeeignet, um die Wirklichkeit abzubilden. Im richtigen Leben gibt es fast immer Ausnahmen. Vielleicht manchmal nur sehr wenige, aber es gibt es gibt sie fast immer.

Ein weiterer Aspekt, der in Betracht zu ziehen wäre, ist Folgender: an einer 1-n-Beziehung kann man nicht (sauber) Attribute für die Beziehung an anhängen (also nicht Attribute der Entitäten, die verbunden werden). Im Gegensatz dazu ist das saubere Anhängen von Attributen an die Beziehung bei m-zu-n-Beziehungen kein Problem, weil ja eine eigene Tabelle verwendet wird.

Fachliche 1-zu-n-Beziehungen (diejenigen, die man im Fachkonzept liest) sind fast immer nur „fast-1-zu-n-Beziehungen“, also „1-zu-n, bis auf ein paar Ausnahmen“. Das darf man aber nicht übersetzen in eine technische 1-zu-n-Beziehung, sondern es bedeutet automatisch: wenn es Ausnahmen gibt, und seinen es auch nur sehr wenige, dann müssen sie eben berücksichtigt werden. Und das heißt: technisch ist es keine 1-zu-n-Beziehung, sondern eine m-zu-n-Beziehung. Und wenn man im Zweifel ist, und lediglich *vermutet*, dass es Ausnahmen gibt, auch wenn man diese gerade nicht konkret benennen kann, so ist es ebenfalls zu empfehlen, zur Vorsicht eine m-zu-n-Beziehung vorzusehen.

Nun könnte man einwenden, auf diese paar Ausnahmen käme es ja gar nicht an. Für die würde sich dann eben schon noch ein geeignetes Verfahren finden. Das ist aber ein sehr gefährlicher Ansatz:

- Denn erstens muss dann, mit Verlaub, schon die Frage erlaubt sein: „Soso, und wer findet dann ‚zu gegebener Zeit‘ ein Verfahren für die Ausnahmen, so nebenbei? Das kommt ja schließlich nicht einfach so vom Himmel. Wie lange werden wir dann Zeit haben, dieses Verfahren zu finden? Und zu implementieren? Und zu testen? Und von welchem Budget wird das bezahlt? Oder sind wir hiermit pauschal von der Pflicht befreit, uns um die Ausnahmen ebenfalls zu kümmern?“ – es dürfte klar geworden sein, was ich meine.

---

1 Leider kann ich dazu keine Quelle mehr finden, weshalb ich für die Korrektheit diese Darstellung nicht einstehen kann. Aber nehmen wir's wenigstens als hübsche, glaubwürdige Anekdote, die sich so ähnlich tatsächlich abgespielt haben könnte.

- Und zweitens können Prozentsätze trügerisch sein: Es mag ja sein, dass die Ausnahmen nur 0,5% aller Daten ausmachen, aber 0,5% können immer noch sehr viel sein, wenn man es mit entsprechenden Datenmengen zu tun hat.

Wenn Sie, wie ich, lange im Bankenumfeld gearbeitet haben, dann wissen Sie, dass z.B. ein paar Milliarden Buchungen am Tag nichts Besonderes sind. 0,5% von, sagen wir, 5 Milliarden sind 25 Millionen. Lassen Sie sich das mal auf der Zunge zergehen, diese „paar Ausnahmen“.

Anzunehmen, dass man das dann schon fallweise und manuell regeln könne, ist schlicht idiotisch.

- Drittens: Selbst, wenn's wirklich nur ganz, ganz wenige Ausnahmen sind, auch absolut gesehen, so ist dennoch nicht gesagt, dass diese Ausnahmen erst dann auftreten, „wenn's gerade passt“, also wenn z.B. das System bereits seit einem Jahr im produktiven Betrieb ist und man bis dahin also eine „Atempause“ gehabt hat.

Im Gegenteil: typischerweise hat man es mit Migrationen von Altdaten ins neue System zu tun, und diese Altdaten enthalten mit Sicherheit bereits Ausnahmen, und die Migration findet *ganz am Anfang* des produktiven Betriebs statt. Zu einer Zeit also, in der man wirklich andere Sorgen hat, als sich schnell mal ein praktikables Verfahren für Sonderfälle auszudenken.

Kurz: Die Ausrede „das sind ja nur Ausnahmen“ gilt in aller Regel nicht.

Zugegeben: Es mag trotzdem immer noch Fälle geben, in denen tatsächlich das Berücksichtigen aller Ausnahmen einen Aufwand verursacht, der nicht mehr zu rechtfertigen ist. Auch die o.g. Regel hat natürlich ihre Ausnahmen. Aber deswegen sage ich ja auch: *fast* immer. Von ganz wenigen, wohlbegründeten Ausnahmen mal abgesehen. Mein Appell ist: Wenn Sie eine 1-zu-n-Beziehung definieren, dann überlegen Sie lieber noch ein zweites und vielleicht noch ein drittes Mal, ob es sich wirklich um eine lupenreine, echte und strenge 1-zu-n-Beziehung ohne Ausnahme handelt.

Wenn Sie geübt sind im Formulieren von Anforderungs-Katalogen, Pflichtenheften und Fachkonzepten, dann kennen Sie vermutlich die Grundregel, nach der man Ausschau halten sollte nach den „Allquantoren“, also den Aussagen, die Worte wie „immer“, „nie“, „jedes“ usw. enthalten – diese sind gefährlich und immer zu hinterfragen, weil es praktisch immer Ausnahmen gibt; und pauschal die Verantwortung für alle, wirklich *alle* Fälle zu übernehmen, ohne Ausnahme, kann sehr teuer werden. In diesem Sinne ist die Aussage „das ist eine 1-zu-n-Beziehung“ für mich ebenfalls ein Allquantor, die genau geprüft werden muss.

Und natürlich gilt dieses Prinzip auch allgemeiner, denn die 1-zu-n-Beziehung ist nur ein besonders häufig anzutreffender Vertreter typischer Fehler in der Datenmodellierung. Es hilft nichts: ein Datenmodell will wohlüberlegt sein. *Jeder* Aspekt muss genau und mehrfach geprüft worden sein. Bloßes, oberflächliches Nachfragen genügt da nicht, man muss sich die Zeit nehmen und richtig tief in die Materie gehen. Und man muss dem Fachbereich viele vermeintlich dumme Fragen stellen. Und sei es auch nur, um sich abzusichern. Ich bin in meinen Projekten mehr als einmal meinen Gesprächspartnern – bewusst – auf die Nerven gegangen mit Fragen wie „Sie sagen also, das ist so und so?“ „Ja.“ „Immer?“ „Ja.“ „Ohne Ausnahme?“ „Ja. (Pause) Es sei denn, natürlich...“ (aha, jetzt kommt's also heraus). Und sehr oft kam dann wirklich noch eine wichtige Information. Deshalb müssen Sie hinterfragen, hinterfragen, hinterfragen. Auch, wenn's nervt. Das sind Sie sich, dem Projektteam, Ihrem Arbeitgeber / Auftraggeber und Ihrem Kunden schuldig.

Richtig, das ist teuer. Aber es ist in aller Regel immer noch billiger, als hinterher, wenn das System bereits im Einsatz ist, noch nachträglich Sachen einzubauen. Genau so wie das Testen.

*Misstrauen Sie grundsätzlich jeder einfachen Aussage!* Wenn Ihr Ansprechpartner vom Fachbereich etwa sagt: „das ist alles gar nicht so kompliziert, und zwar...“, dann sollte das für Sie ein Alarmzeichen sein: Schlecht vorbereitete und „flott“ angegebene Anforderungen taugen nur für einen ersten Wurf – des Datenmodells, nicht des produktiven Systems! Meistens ist es dann doch ein bisschen komplizierter. Das Geschäft ist nicht einfach, warum sollte es also das Datenmodell sein?