

**Praktikum Datenbanken / DB2**  
**Woche 3: Relationenschemata, 1. Normalform**

Raum: LF 230

Nächste Sitzung: 27./30. Oktober 2003

Aktuelle Informationen unter:

[http://www.is.informatik.uni-duisburg.de/teaching/lectures/dbp\\_ws03/index.html](http://www.is.informatik.uni-duisburg.de/teaching/lectures/dbp_ws03/index.html)

## E-R-Modell zu Relationenschema

Als zweiter Schritt beim Entwurf einer Datenbank steht nach der Modellierung die Umsetzung des Modells in ein Relationenschema an. Ziel soll ein Schema sein, das möglichst direkt in einer konkreten relationalen Datenbank implementiert werden kann.

Im Allgemeinen gibt es bei dieser Umsetzung mehrere geeignete Schemavarianten. Eine erste Annäherung für die Umsetzung kann wie folgt aussehen:

- ein Entitätentyp wird in eine Relation mit den gleichen Attributen wie der Entitätentyp überführt
- eine Beziehung wird in eine Relation überführt, deren Attribute die Schlüssel der von ihr verbundenen Relationen sind

Normalerweise gibt es allerdings noch einige andere Dinge zu beachten. Im Folgenden werden ein paar einfache Faustregeln dazu angegeben. Dabei ist zu bedenken, dass in Hinblick auf eine konkrete Anwendung es manchmal sinnvoll sein kann, eine theoretisch nicht optimale Umsetzung zu wählen.

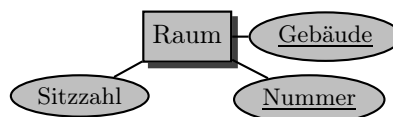
Relationenschemata werden wie in **Kemper/Eick** beschrieben spezifiziert:

$$\text{SchemaName} : \{ [ \underline{\text{Attribut}_1} : \text{Typ}_1, \text{Attribut}_2 : \text{Typ}_2, \dots ] \}$$

Dabei wird der Primärschlüssel einer Relation durch Unterstreichung gekennzeichnet. Attribute innerhalb einer Relation müssen eindeutig benannt sein.

### Entitätentypen

Ein Entitätentyp wird normalerweise als Relation abgebildet, deren Attribute die Attribute dieses Typs sind. Diese Relation hat zunächst einmal keinerlei Verbindung zu den Beziehungen, an denen der Entitätentyp teilnahm. Mehrwertige Attribute können in ein mengen- oder listenwertiges Attribut abgebildet werden. Diese werden dann bei der Normalisierung behandelt.


$$\text{Räume} : \{ [\underline{\text{Nummer}} : \text{integer}, \underline{\text{Gebäude}} : \text{string}, \text{Sitzzahl} : \text{integer}] \}$$

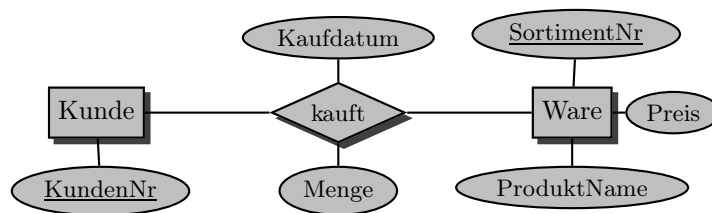
Zusammengesetzte Attribute werden entweder aufgelöst, d.h. die Attribute des zusammengesetzten Attributes werden zu Attributen der Umsetzung des Entitätstyps, oder sie werden als eigene Relation aufgefasst. Diese neue Relation enthält alle Attribute des zusammengesetzten Attributes und alle Primärschlüsselattribute der Ursprungsrelation. Diese Fremdschlüssel bilden den Primärschlüssel der neuen Relation.

## Beziehungstypen

Ein Beziehungstyp wird üblicherweise als eine eigene Relation abgebildet. Als Attribute für diese Relation werden herangezogen:

- die Attribute des Beziehungstyps
- die Primärschlüsselattribute der teilnehmenden Entitätstypen als Fremdschlüssel

Der Primärschlüssel der Relation enthält die Fremdschlüssel und eventuell zusätzliche Attribute.



kauft : {[KundenNr: integer, SortimentNr: integer,  
Kaufdatum: date, Menge: integer]}

Nimmt ein Entitätentyp mehrfach an einer Beziehung teil, so müssen seine Schlüsselattribute entsprechend oft in die Relation übernommen werden, die dieser Beziehung entspricht. Dabei muss man die Attribute umbenennen, um Namenskonflikte zu vermeiden.

## Schwache Entitätstypen

Ein schwacher Entitätstyp wird als eine eigene Relation abgebildet, die zusätzlich zu den eigenen Attributen auch den Primärschlüssel des Vater-Entitätstypen als Fremdschlüssel enthält. Dieser bildet zusammen mit den ausgezeichneten Teilschlüsseln des schwachen Entitätstypen den Primärschlüssel der Relation.

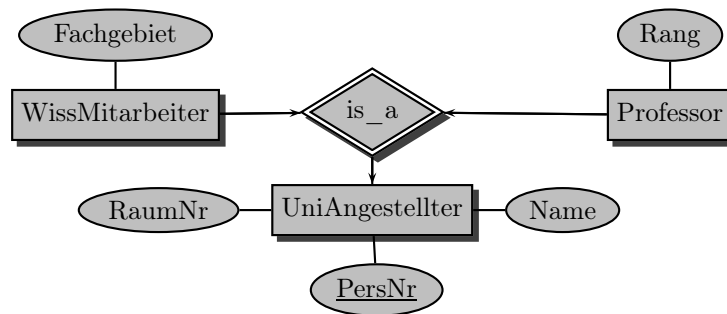
Die Beziehung zwischen einem schwachen Entitätstypen und dem Vater-Entitätstyp muss im Allgemeinen überhaupt nicht nachgebildet werden.

## Generalisierung

Generalisierung wird durch das relationale Modell nicht direkt unterstützt. Das relationale Modell verfügt über keine Vererbung. Man kann aber versuchen, Generalisierung mit den existierenden Mitteln nachzumodellieren. Dafür gibt

es unterschiedliche Strategien, wobei sich die folgende am engsten an die E-R-Modellierung hält:

- Für jeden Entitätstypen E in der Generalisierungshierarchie gibt es eine Relation mit den Schlüsselattributen des Obertypen und den Attributen von E



UniAngestellte : {[PersNr, Name, RaumNr]}

Professoren : {[PersNr, Rang]}

WissMitarbeiter : {[PersNr, Fachgebiet]}

Eine Alternative wäre die Überführung des abgeleiteten Entitätstypen E in eine Relation, die als Attribute alle Attribute des Obertypen sowie die Attribute von E besitzt. In dieser Variante würden in der Relation des Obertyps nur Instanzen gespeichert, die zu keinem der Subtypen gehören.

UniAngestellte : {[PersNr, Name, RaumNr]}

Professoren : {[PersNr, Name, RaumNr, Rang]}

WissMitarbeiter : {[PersNr, Name, RaumNr, Fachgebiet]}

## Zusammenführen von Relationen

Die direkte Überführung wie oben beschrieben liefert oft nicht die bestmöglichen Relationen. Oft kann man nun noch Relationen zusammenführen. Allgemein gilt: Relationen mit dem gleichen Schlüssel kann man zusammenfassen (aber auch nur diese).

Ein binärer Beziehungstyp R zwischen Entitätentypen E und F, an dem mindestens einer der beiden beteiligten Entitätstypen (sagen wir E) mit Funktionalität 1 teilnimmt, kann mit der E entsprechenden Relation zusammengeführt werden. Dazu führt man eine Relation mit folgenden Attributen ein:

- die Attribute von E
- die Primärschlüsselattribute von F
- die Attribute der Beziehung R

Überlegt selbst, warum das vorteilhafter ist, als die Überführung der Beziehung R in eine eigene Relation. Überlegt auch, was es für Gegenargumente gibt.

## Normalformen

Normalformen stellen eine theoretische Basis dar, um relationale Datenbankschemata bewerten zu können. Insbesondere sollen durch Normalisierung Redundanzen und implizite Darstellung von Information verhindert werden. Durch Einhalten der Normalformen bei der Modellierung können bei der späteren Benutzung Änderungs-, Einfüge- oder Löschanomalien verhindert werden.

Normalformen werden zu einem späteren Zeitpunkt in der Vorlesung Datenbanken behandelt. Hier soll nur die erste Normalform kurz erklärt werden. Über die schärferen Normalformen kann man sich in der weiterführenden Literatur informieren (die Literaturliste findet sich auf dem 1. Arbeitsblatt).

### Erste Normalform

Eine Relation ist in *Erster Normalform (1NF)*, wenn alle Attribute nur atomare Werte annehmen dürfen. Zusammengesetzte oder mengenwertige Attribute sind unter der 1NF nicht zulässig. In klassischen relationalen DBMS lassen sich Relationen, die nicht in 1NF sind, nicht speichern. Die Relation

Eltern : {[VaterName, MutterName, {KindName}]}

ist nicht in 1NF.

Nachteile einer Relation, die nicht in 1NF ist, sind unter anderem fehlende Symmetrie, Speicherung redundanter Daten und Aktualisierungsanomalien. Bei der Überführung in 1NF entstehende neue Redundanzen werden im Laufe der weiteren Normalisierung beseitigt.

Vorgehen:

- Listen- bzw. mengenwertige Attribute werden durch ihre Elemente ersetzt. Auf Tupelebene wird das durch Ausmultiplikation der Liste mit dem restlichen Tupel erreicht.
- Zusammengesetzte Attribute werden durch die Teilattribute ersetzt.
- Alternativ können auch neue Relationen gebildet werden.

## Aufgaben

- (a) Vervollständigt die Modellierung aus der letzten Woche und ergänzt sie um die Modellierung des DVD-Versands.
- (b) Nehmt Eure Modellierung der Mini-Welt der Filme und Serien, und versucht, diese zunächst möglichst modellierungsgetreu in ein relationales Schema zu übertragen.

- (c) Bringt das relationale Schema in 1. Normalform. Verfeinert dann das relationale Schema soweit möglich durch Elimination (Zusammenführung) von Relationen.
- (d) Arbeitet Probleme Eurer Modellierung heraus und versucht Euren ursprünglichen Entwurf zu verbessern. Vergleicht den Entwurf dazu auch mit der Musterlösung, und dokumentiert Eure Änderungen.