

## Übung zur Vorlesung „Datenbanksysteme für Hörer anderer Fachrichtungen“

Richard Kuntschke (richard.kuntschke@in.tum.de)

### Lösungen zu Blatt 2

#### Aufgabe 1

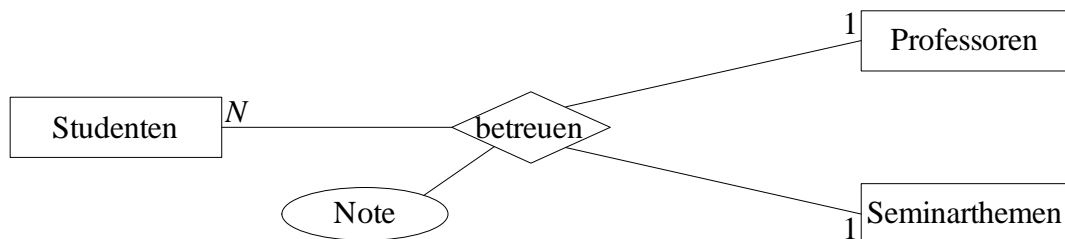


Abbildung 1: Modellierung der Betreuung von Seminarthemen

In Abbildung 1 ist die dreistellige Beziehung *betreuen* dargestellt, welche die folgenden partiellen Funktionen definiert:

$$\begin{aligned} \textit{betreuen} &: \textit{Professoren} \times \textit{Studenten} \rightarrow \textit{Seminarthemen} \\ \textit{betreuen} &: \textit{Seminarthemen} \times \textit{Studenten} \rightarrow \textit{Professoren} \end{aligned}$$

Damit sind folgende Konsistenzbedingungen festgelegt:

- Studenten dürfen bei einem Professor nur **ein** Seminarthema bearbeiten.
- Studenten dürfen dasselbe Thema nur bei **einem** Professor bearbeiten.

Ein Professor, kann dasselbe Seminarthema (etwa in unterschiedlichen Jahrgängen) allerdings an unterschiedliche Studenten vergeben. In der Relation *betreuen* kann ein und derselbe Student in unterschiedlichen Tupeln auftreten. Dasselbe trifft auf Professoren und Seminarthemen zu. Dadurch ergibt sich, dass die Multiplizitäten in der  $(min, max)$ -Notation jeweils  $(0, *)$  sein müssen. Damit können jedoch die geforderten Konsistenzbedingungen verletzt werden. Im Speziellen ist es dann möglich, dass ein Student bei einem Professor zweimal dasselbe Thema bearbeitet.

In Abbildung 2 wird mittels der Relation *betreuen* die Betreuung von Diplomarbeiten modelliert. Die Kardinalitäten sind in der  $(min, max)$ -Notation dargestellt. Da dasselbe Diplomarbeitsthema von verschiedenen Professoren mehrfach vergeben werden kann und ein Professor mehrere Diplomanden betreut, haben *DAThemen* und *Professoren* jeweils die Kardinalität  $(0, *)$ . Ein Student schreibt allerdings höchstens eine Diplomarbeit, somit ergibt sich eine Kardinalität von  $(0, 1)$ . Mittels Funktionalitätsschreibweise lässt sich dieser Sachverhalt jedoch nicht beschreiben. Da für einen Studenten automatisch festgelegt ist, bei welchem Professor er welches Thema hat, wäre die Funktionalität 1:1:1.

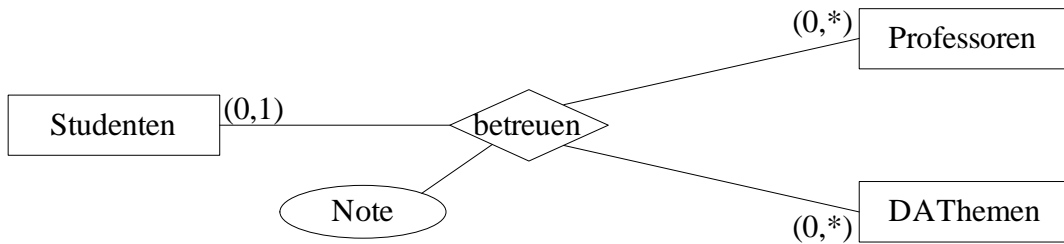


Abbildung 2: Modellierung der Betreuung von Diplomarbeiten

### Aufgabe 2

Eine mögliche Modellierung ist in Abbildung 3 dargestellt. Folgende Eigenschaften werden hier umgesetzt:

- Jeder Bahnhof liegt in einer Stadt. Eine Stadt kann aber wiederum mehrere Bahnhöfe haben.
- Jeder Zug hat einen ausgewiesenen Start- und Zielbahnhof, d. h. fährt auf einer festen Route.
- Auf dieser Route können mehrere Zwischenstationen auftreten. Dies wird über die dreistellige Relation *verbindet* realisiert. Die Ankunfts- und Abfahrtszeiten stellen Attribute der Relation dar.

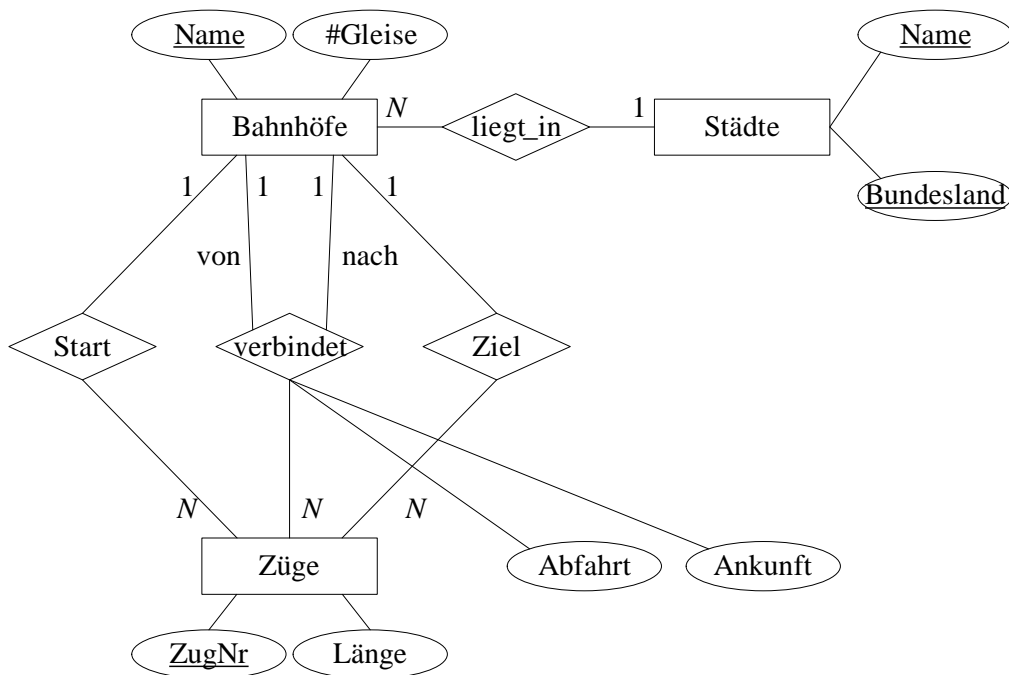


Abbildung 3: Modellierung eines Zugauskunftsystems

### Aufgabe 3

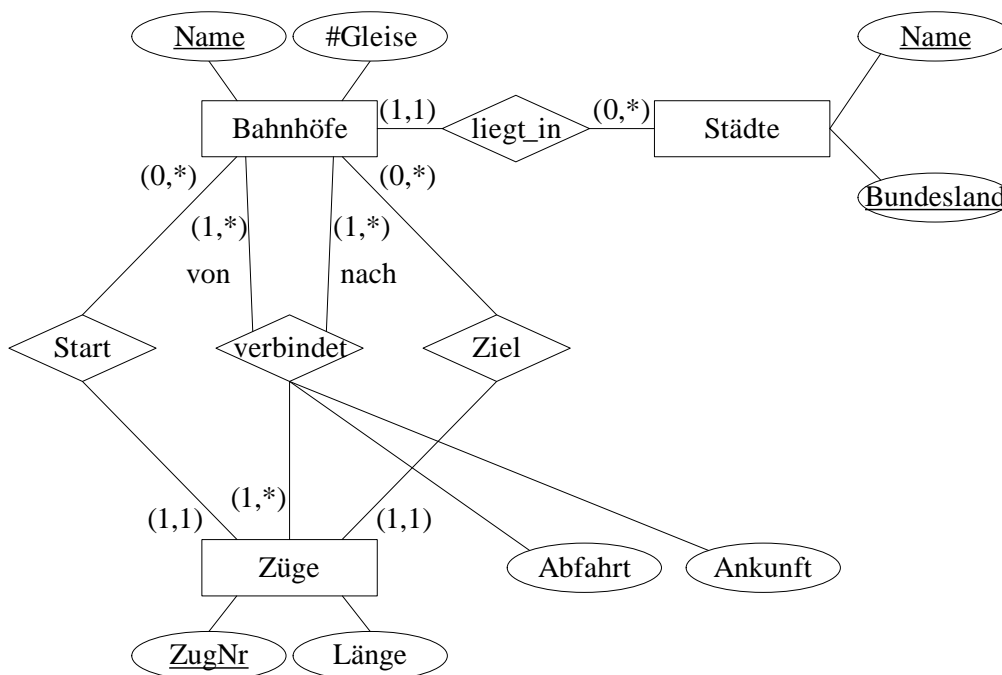


Abbildung 4: ER-Modellierung von Zugverbindungen mit Angabe der Kardinalitäten in  $(min, max)$ -Notation

Initiale Überführung in das relationale Modell:

Relationen aus Entitytypen:

$$\text{Städte} : \{ \{ \underline{\text{Name}} : \text{string}, \text{Bundesland} : \text{string} \} \} \quad (1)$$

$$\text{Bahnhöfe} : \{ \{ \underline{\text{Name}} : \text{string}, \# \text{Gleise} : \text{integer} \} \} \quad (2)$$

$$\text{Züge} : \{ \{ \underline{\text{ZugNr}} : \text{integer}, \text{Länge} : \text{integer} \} \} \quad (3)$$

Relationen aus Beziehungstypen:

$$\text{liegt\_in} : \{ \{ \underline{\text{BName}} : \text{string}, \text{SName} : \text{string}, \text{Bundesland} : \text{string} \} \} \quad (4)$$

$$\text{Start} : \{ \{ \underline{\text{ZugNr}} : \text{integer}, \text{BName} : \text{string} \} \} \quad (5)$$

$$\text{Ziel} : \{ \{ \underline{\text{ZugNr}} : \text{integer}, \text{BName} : \text{string} \} \} \quad (6)$$

$$\text{verbindet} : \{ \{ \underline{\text{VonBahnhof}} : \text{string}, \text{NachBahnhof} : \text{string} \} \} \quad (7)$$

$$: \{ \{ \underline{\text{ZugNr}} : \text{integer}, \text{Abfahrt} : \text{date}, \text{Ankunft} : \text{date} \} \} \quad (8)$$

Verfeinerung des Schemas durch Zusammenfassen von Relationen:

Relationen für binäre Beziehungstypen werden mit Relationen für Entitytypen zusammengefaßt, falls diese gleiche Schlüssel besitzen und es sich dabei um 1 : N-, N : 1- oder 1 : 1-Beziehungen handelt.

$\Rightarrow (4) \mapsto (2), (5) \mapsto (3), (6) \mapsto (3)$

Resultierendes Schema:

Städte : {[Name : string, Bundesland : string]}

Bahnhöfe : {[Name : string, #Gleise : integer,  
: SName : string, Bundesland : string]}

Züge : {[ZugNr : integer, Länge : integer]}

          : StartBahnhof : string, Zielbahnhof : string]}

verbindet : {[VonBahnhof : string, NachBahnhof : string  
: ZugNr : integer, Abfahrt : date, Ankunft : date]}

Im vorliegenden Fall ist die Zugnummer eindeutig für eine Verbindung. Ein ICE, der die Städte München (Startbahnhof) und Berlin (Zielbahnhof) an einem bestimmten Tag verbindet, hat somit eine eindeutige Zugnummer für diese Verbindung, die über mehrere Zwischenbahnhöfe erfolgen kann. Fährt der Zug zurück, erhält er eine andere Nummer zugewiesen. Dadurch sind die Kombinationen (*ZugNr*, *VonBahnhof*) und (*ZugNr*, *NachBahnhof*) zwei mögliche Schlüssel für die Relation *verbindet*.