Prof. DI Dr. Erich Gams

Relationenmodell

Begriffe, Grundlagen, Überführung ERM-RM

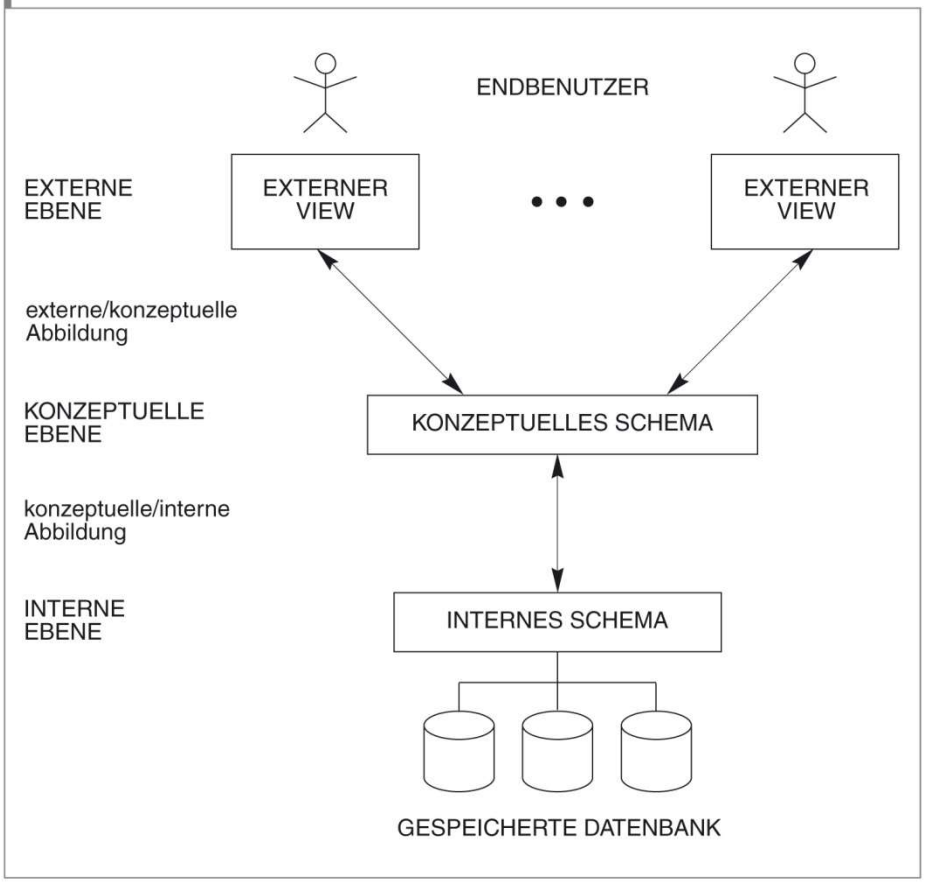
informationssysteme htl-wels

Übersicht ☛ Was lernen wir?

- Einordnung Datenmodelle
- Das relationale Modell (Begriffe, Grundlagen)
- Schlüssel
  - Übungen
- Vergleich EM-RM
- Transformationsregeln
  - Basisregeln
- Übungen

Datenbankarchitektur

Abbildung 2.2: Darstellung der Drei-Schichten-Architektur.



Entwurf von Datenschemas

- Beim Entwurf einer Datenbank wird
- zwischen 2 Abstraktionsstufen und
- ihren entsprechenden Datenschemas
  - konzeptionelles Datenschema und logisches Datenschema
- unterschieden.

Datenschemas

- Konzeptionelles Datenschema:
  - ist eine systemunabhängige Datenbeschreibung,
  - d.h. sie ist unabhängig von den eingesetzten Datenbank- und Computersystemen. (ZEHNDER 1998)

  ⇒ ER-Modell

- Logisches Datenschema:
  - beschreibt die Daten in der Datenbeschreibungssprache (DDL = Data Definition Language) eines bestimmten Datenbank-Verwaltungssystems. (ZEHNDER 1998

  ⇒ Relationales Datenmodell

Logische Datenmodellierung

- Im Gegensatz zum Entity-Relationship-Modell (ER-Modell oder ERM) liegt das Relationenmodell in gewisser Hinsicht eine Stufe „tiefer“
- **Ziele**
  - das Anordnen der zu speichernden Informationen mit Hilfe eines Modells,
  - das eine möglichst redundanzfreie Speicherung unterstützt und geeignete Operationen für die Datenmanipulation zur Verfügung stellt. (gitta.info)

Unterschiede Datenmodelle

- Das ER-Modell ist ein *unvollständiges Datenmodell*.
  - Es beschreibt nur die Daten.

- Implementiert wird das ER-Modell dann in einem „*vollständigen“ Datenmodell* (Relationales Modell), d.h. es hat:
  - eine Beschreibung der Daten
  - mit zugehörigen Operationen, um mit den Daten zu arbeiten.

Überblick: Relationales Modell

- Modell für Daten und Operationen auf diesen Daten.

- Fundierte mathematische Grundlage:
  - die Mengentheorie
- Einfaches Datenmodell:
  - Grundstruktur Relation
  - Tuple, Domain, Attribute, Key Attribute.
- Einfache Operationen:
  - Definition von Relationen
  - Eingabe und Änderung von Daten
  - Suche nach Daten.
- In relationalen Datenbanksystemen implementiert
  - (Oracle, DB2, SQL Server, MySQL , Access...)

Begriff: Domäne

Definition:

Zu jedem Attribut A_i, $1 \leq i \leq n$, gibt es eine Menge D_i, den ***Wertebereich*** (domain) von A_i.

Notation:

$dom(A_i)$ ist der Wertebereich von A_i.

Beispiel:

Das Attribut **GESCHLECHT** hat den Wertebereich ***dom(GESCHLECHT) = {männlich, weiblich}***.

Grundlagen des relationalen Modells

Seien D_1, D_2, ..., D_n Domänen (=Wertebereiche)

- Relation: ist das kartesische Produkt (Kreuzprodukt) der *n* Domänen

  $R \subseteq D_1$ x ... x D_n Relation

  Bsp.: *Telefonbuch* $\subseteq$ *string x string x integer*

- Tupel: Element der Menge R

  $t \in R$

  Bsp.: *t* = („Mickey Mouse“, „Main Street“, 4711)

- Relationenschema: legt die Struktur der gespeicherten Daten fest (abstrakte Beschreibung einer Relation)

  Bsp.:

  Telefonbuch: {[*Name: string, Adresse: string, Telefon#:integer*]}

Relationale Datenbank

- Relationales Datenbankschema:
  - Ein relationales Datenbankschema ist eine Menge von Relationenschemata S = {R1, ..., Rn}

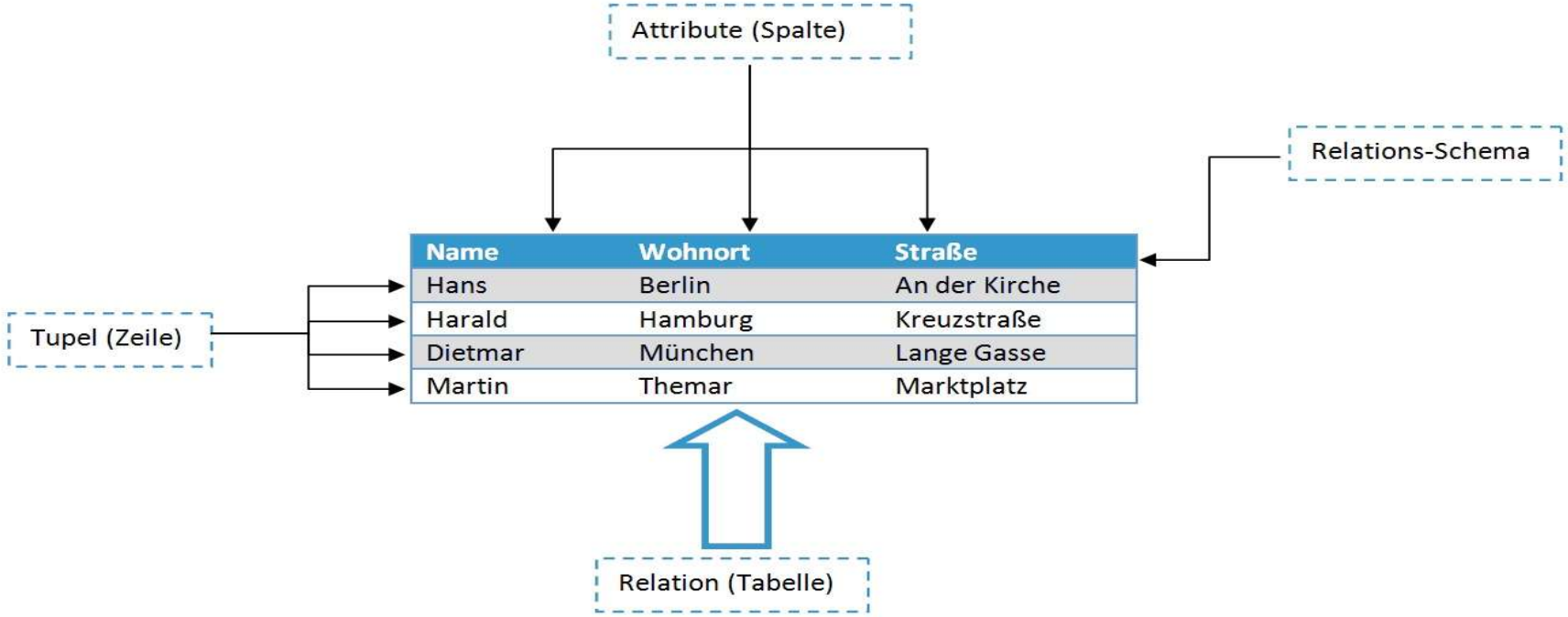
Beispiel:

Person: {[Vorname: string, Nachname: string, Adresse: string, Pers#:integer]}

Firma: {[Firmen#:integer, Name:string, Adresse:string]}

Eine relationale Datenbank ist ein **relationales Datenbankschema** mit einer entsprechenden Datenbankinstanz.

Tabelle: Relation mit Tupeln



Name	Wohnort	Straße
Hans	Berlin	An der Kirche
Harald	Hamburg	Kreuzstraße
Dietmar	München	Lange Gasse
Martin	Themar	Marktplatz

- Keine Duplikate:
  - Jedes Tupel kommt mit seinen Daten nur einmal vor
- Keine Ordnung:
  - Reihenfolge der Tupel beliebig und zufällig

Vergleich der Konzepte von ERM und RM

ERM	RM
Entity Type	Relation
Relationship Type	Relation
Entity	Tupel
Relationship	Tupel
Attribut	Attribute
mehrwertiges Attribut	?? Attribute abwandeln
Schlüssel	Schlüssel
Generalisierung/ Spezialisierung	??
Totalität einer Beziehung (muss, kann)	?? -> Erweiterung, NOT NULL, FOREGN KEY
Kardinalität einer Beziehung (1:N, N:M, 1:1)	??

Attribute und Schlüssel

Schlüssel

- Die Identifikation und Verknüpfung von Datenbeständen erfolgt über einen Schlüssel.
- Ein Schlüssel muss ein Tupel eindeutig identifizieren.
  - Die Werte zweier Schlüsselpaare dürfen nicht identisch sein
  - De Schlüsselattribute dürfen keine undefinierten Werte (Nullwerte) enthalten

Attribute und Schlüssel

Schlüsselkandidat

- Der Schlüsselkandidat kann aus einem Attribut oder einer Menge von Attributen bestehen, wobei er eindeutig und minimal sein muss.
  - d.h. durch Weglassen eines Attributes darf kein kleinerer Schlüsselkandidat entstehen
  - Es darf ein künstliches Attribut als Schlüsselkandidat eingeführt werden.
  - Es kann mehrere Schlüsselkandidaten geben!

Primärschlüssel (Identifikationsschlüssel)

- Der für den Anwendungsfall zur Identifikation tatsächlich ausgewählte Schlüsselkandidat.
- Während der gesamten Existenz eines Objekts in einer Datenbank darf der Primärschlüssel nicht verändert werden.
- Englisch: Primary-Key
- Wird unterstrichen

Eine **minimale**(!) Menge von Attributen, welche das zugeordnete Entity eindeutig innerhalb aller Entities seines Typs identifiziert

Übung

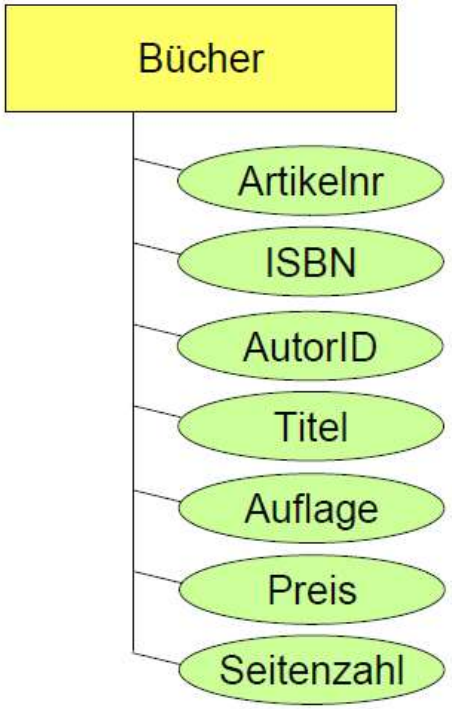
› Was sind **geeignete Schlüsselkandidaten** für diese Beispiele?

raum	ausstattung	plätze	bemerkungen
A111	'Overhead-Projektor'	40	'schmale Tische'
E027	'22 PC'	22	'im Sommer heiss'

titel	jahr	minuten	produktion
Dr. Jekyll & Mr. Hyde	1938	94	Columbia
Dr. Jekyll & Mr. Hyde	1941	110	Universal
Jackie Brown	1998	145	Universal
La Vita e Bella	1998	123	United Artist

Übung

› Bestimme alle **Schlüsselkandidaten** und den **Primärschlüssel**!



Attribute und Schlüssel

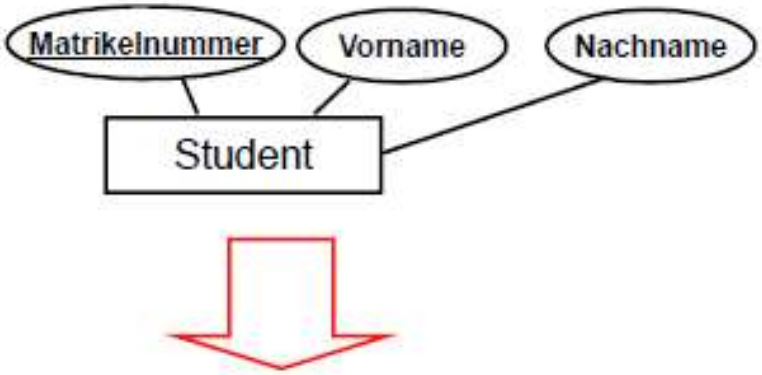
- Eine *saubere* ER-Modellierung läßt sich leicht in ein gutes Relationenmodell überführen.

- Anwendung von Transformationsregeln:
  - **(Basisregeln** erzeugen korrekte Abbildung, aber nicht optimiert bzg. Anzahl Relationen bzw. Datenmengen)
  - (Optimierungsregeln sparen Relationen und Datenmengen)

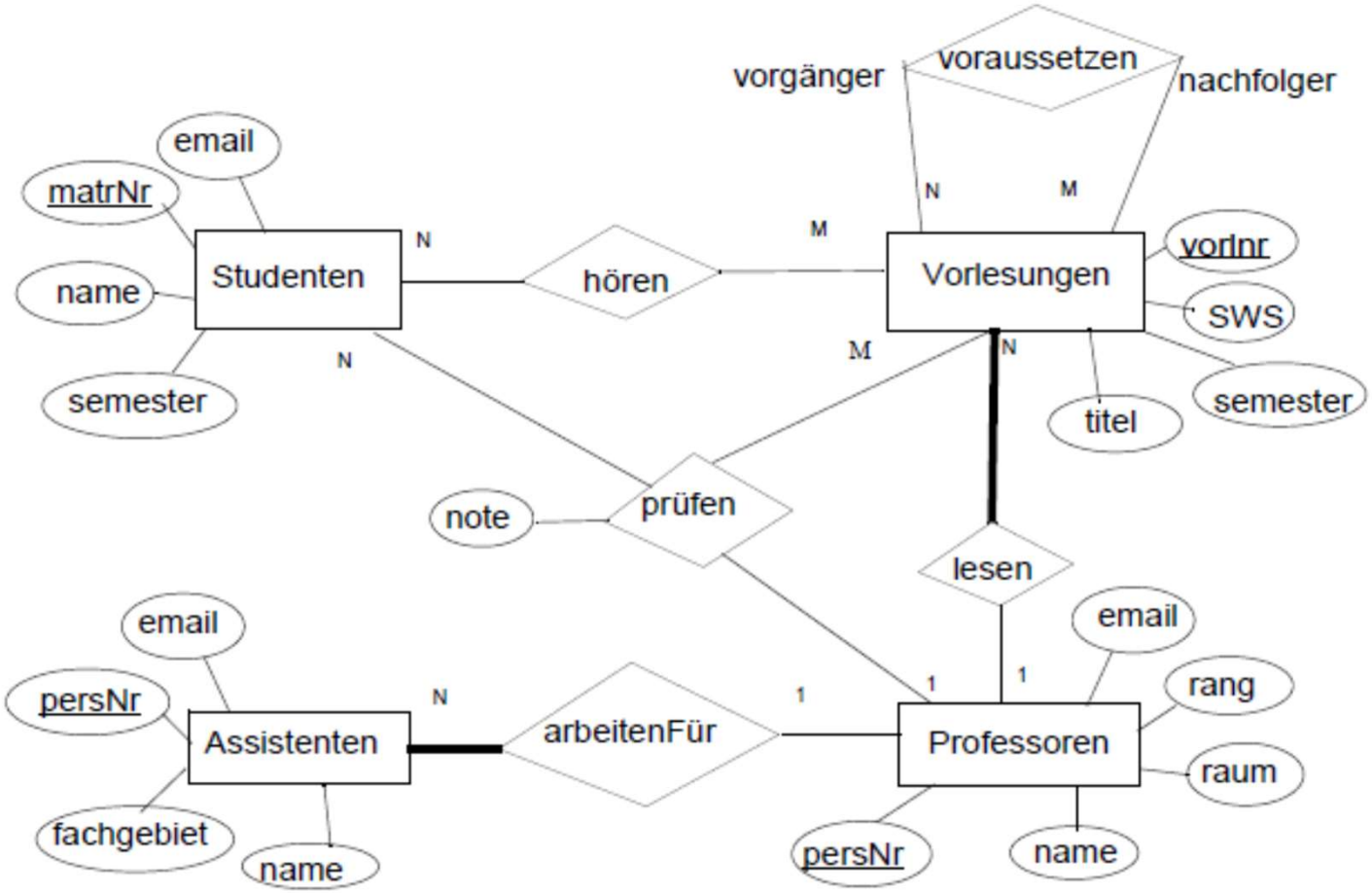
Überführung von Entitätstypen ins Relationenmodell

- **Wandle alle Entity Types in Relationen um!**
  - Schlüssel -> Schlüssel
  - (ER-)Attribute -> (RM-)Attribute

- Beispiel:



Uni-Beispiel



Relationale Darstellung von Entitätstypen

Studenten: {[MatrNr:integer, *Name*: *string, Semester*: *integer*]}

Vorlesungen: {[VorlNr:integer, *Titel*: *string, SWS*: *integer*]}

Professoren: {[PersNr:integer, *Name*: *string, Rang*: *string, Raum*: *integer*]}

Assistenten: {[PersNr:integer, *Name*: *string, Fachgebiet*: *string*]}

Fremdschlüssel

- **Bezieht sich** ein(e) **Attribut(gruppe) einer Relation auf den Primärschlüssel einer anderen,** so wird es (sie) **in der referenzierenden Relation Fremdschlüssel (foreign key)** genannt**.**
- **Fremdschlüssel (strichliert)** dienen zur **Verknüpfung mehrerer Relationen.**
- Beispiel:

Fremdschlüssel

Rechnungen

Rechnungsnummer	Datum	Kundennummer
12454	1.1.2000	3003587
65432	13.5.2000	3003587
87342	24.8.2000	3110020

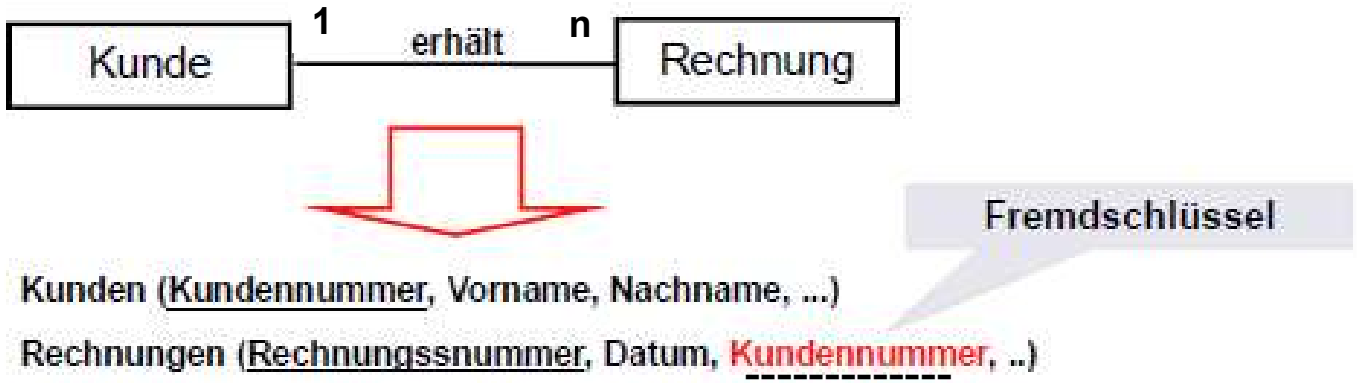
Kunden

Kundennummer	Vorname	Nachname
3003587	Christian	Schulz
3072456	Martin	Seger
3110020	Julia	Maier

- **Kundennummer** ist *Primärschlüssel* in der Relation Kunden und *Fremdschlüssel* in der Relation Rechnungen.

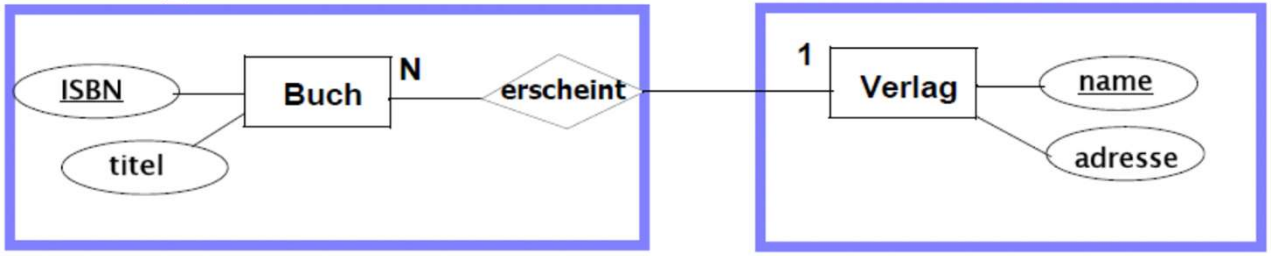
Überführung von 1:n und c:n – Beziehungen

- **Überführung von 1:n – Beziehungen**
  - durch Verknüpfung der Relationstypen mittels Fremdschlüssel im Relationstyp mit der Kardinalität n
- **Überführung von c:n – Beziehungen**
  - durch Verknüpfung der Relationstypen mittels Fremdschlüssel im Relationstyp mit der Kardinalität n

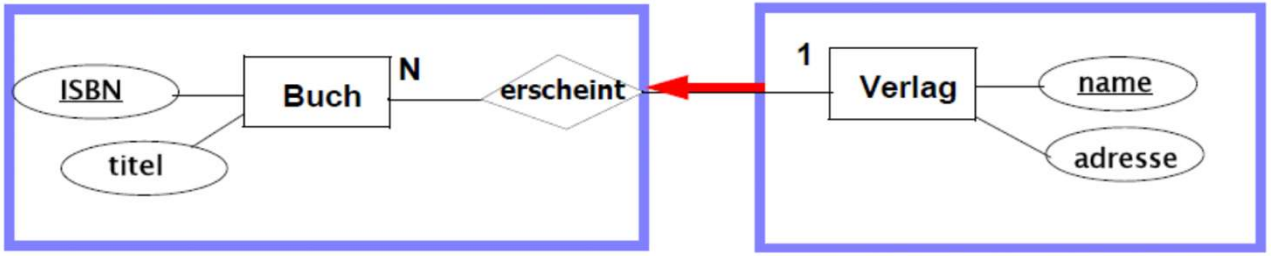


N:1 Beziehungen

- Beziehung auf der Seite des N ansiedeln:



- Schlüssel der anderen Relation als Fremdschlüssel übernehmen:



- Verlag: {[name, adresse]}
- Buch: {[ISBN, titel, **verlag**]}

Transformationsregeln in a nut shell

- Regel 1: Wandle alle Entity Types in Relationen um.
- 1:N
  - Schlüssel des Entity Type auf der 1(oder C) Seite wandert als Fremdschlüssel zu Entity Type auf der N Seite

Überführung von 1:1 Beziehung

1. Phase: Naives Auflösen in 3 Relationen

Diplomarbeit {[titel, abstract, fachbereich]}
Gutachten {[note, datum, text]}
gehört_zu {[titel, datum]}

2. Phase: Zusammenlegen nach gleichen Schlüsseln in 2 Relationen

Diplomarbeit {[titel, abstract, fachbereich, datum]}
Gutachten {[note, datum, text]}

Überführung von 1:1 Beziehung

Eventuell: alles zu einer Relation zusammenlegen

```
Diplomarbeit
{[titel, abstract, fachbereich, datum, note, gutachtenText]}
```

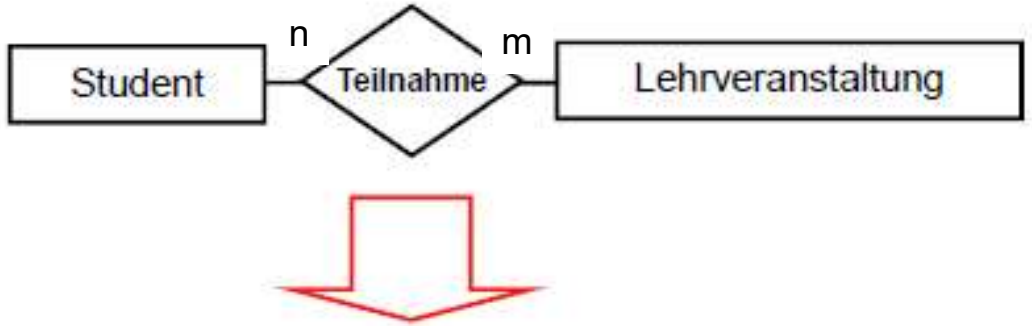
1. Frage: gehören diese Daten wirklich alle immer zusammen, oder "stört" die Kombination?
2. Frage: hat Gutachten vorher Beziehungen zu anderen Entity Types gehabt, müssen die nun bei Gutachten versorgt werden. Ist das immer sinnvoll?

Transformationsregeln in a nut shell

- Regel 1: Wandle alle Entity Types in Relationen um.
- 1:N
  - Schlüssel des Entity Type auf der 1(oder C) Seite wandert als Fremdschlüssel zu Entity Type auf der N Seite
- 1:1
  - Zusammenlegen in 2 Relationen (1 Schlüssel wandert als Fremdschlüssel in die „andere“ Relation)
  - (Eventuell: nur 1 Tabelle)

Überführung von n:m – Beziehungen

- **Es muss ein eigener Relationstyp für den Beziehungstyp gebildet werden.**
- Der Primärschlüssel dieses Relationstyps wird aus den Primärschlüsseln der beiden anderen Relationstypen zusammengesetzt.



Studenten (Matrikelnummer, Vorname, Nachname, ...)

Lehrveranstaltungen (Veranstaltungsnummer, Termin, Raum, ..)

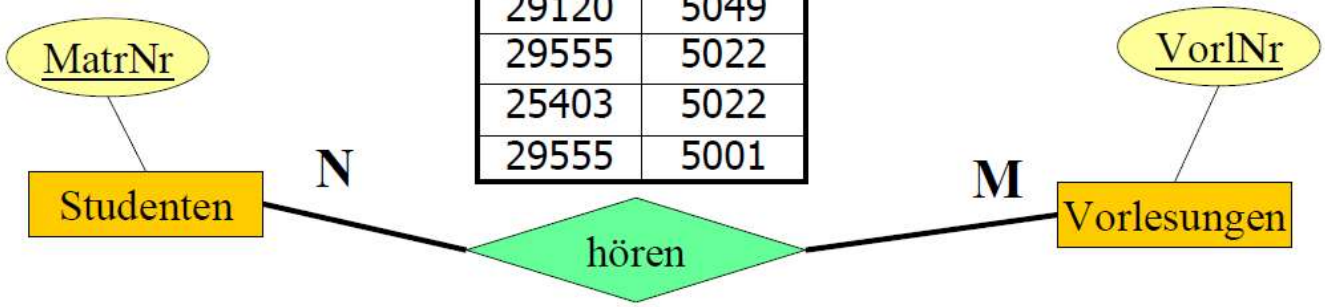
Teilnahme (Matrikelnummer, Veranstaltungsnummer)

Weiteres Beispiel einer N:M Beziehung

Studenten	
MatrNr	...
26120	...
27550	...
...	...

hören	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

Vorlesungen	
VorlNr	...
5001	...
4052	...
...	...



Bezeichnung der Relation (RM) eines Relationship Types (ERM)

- Es ist auch möglich die Relation nach den beiden verknüpften Relationstypen zu benennen.
  - *Studenten_Vorlesung*
  - Dies passiert vor allem, wenn der Relationship Type im ERM nicht bezeichnet ist oder fehlt.
- Relationsnamen können in der Mehrzahl oder Einzahl bezeichnet werden.
  - *Studenten* oder *Student*
- Man sollte sich auf eine Art einigen.

Transformationsregeln in a nut shell

- Wandle alle Entity Types in Relationen um.
- 1:N
  - Primärschlüssel des Entity Type auf der 1(oder C) Seite wandert als Fremdschlüssel zu Entity Type auf der N Seite
- 1:1
  - Zusammenlegen in 2 Relationen (1 Primärschlüssel wandert als Fremdschlüssel in die „andere“ Relation)
  - (Eventuell: nur 1 Tabelle)
- N:M
  - Relationship Type in eigene Tabelle umwandeln, Primärschlüssel beider Entity Types wandern in diese Tabelle

Transformationsbeispiel