

Datenmanagement & -analyse

Datenmodellierung: Logischer Datenbankentwurf mit dem relationalen Modell

Prof. Dr. Christoph M. Flath

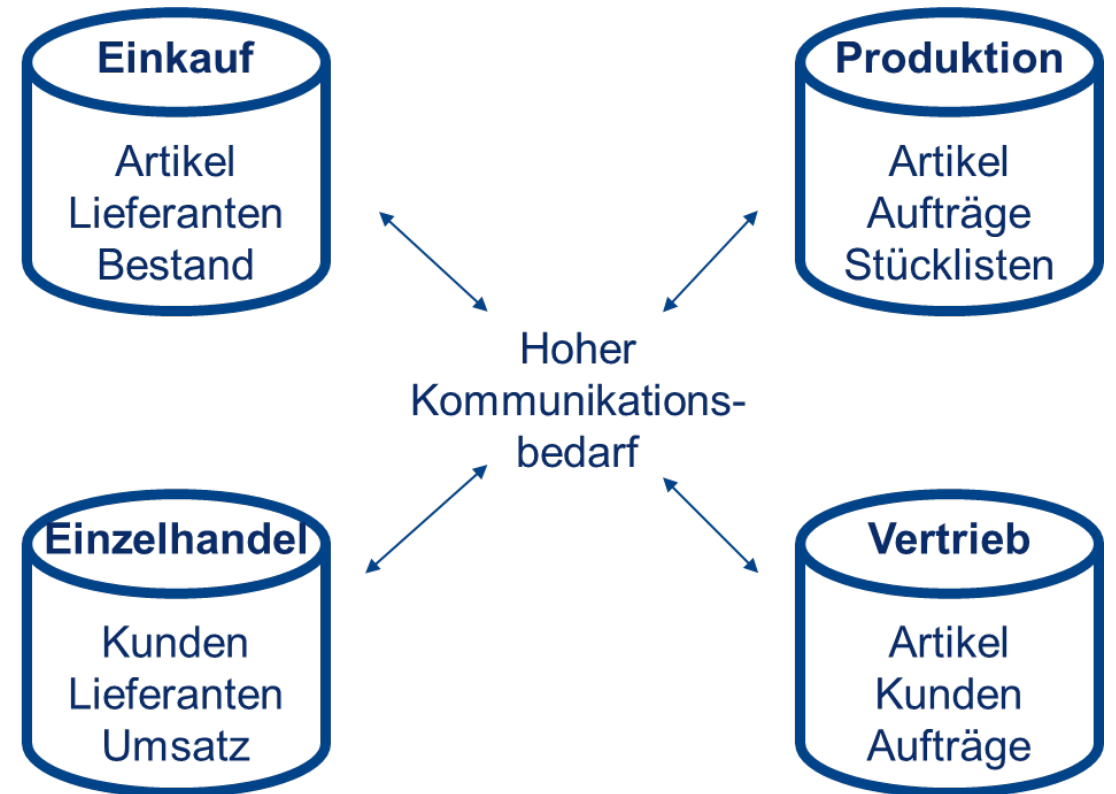
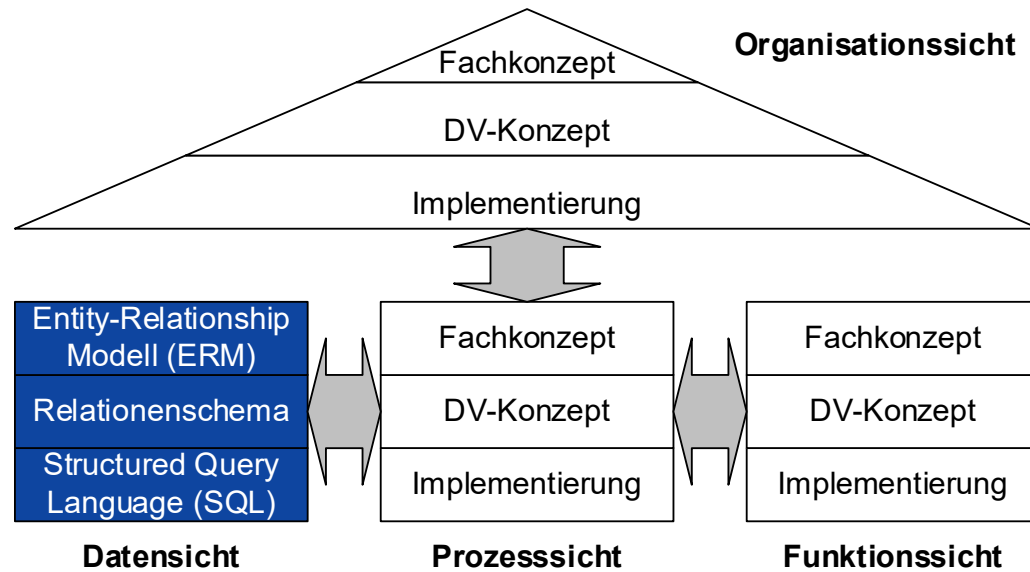
Lehrstuhl für WI & BA

Julius-Maximilians-Universität Würzburg

Sommersemester 2021

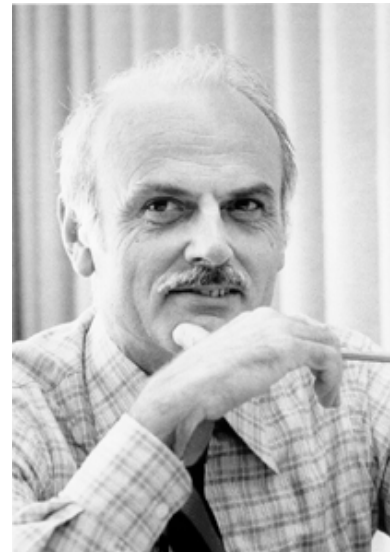


Architektur Integrierter Informationssysteme



Scheer (1992)

- **Edgar Frank „Ted“ Codd:**
A Relational Model of Data for
Large Shared Data Banks.
Communications of the ACM
6 (13) 1970, S. 377-387.



Information Retrieval

P. BAXENDALE, Editor

A Relational Model of Data for Large Shared Data Banks

E. F. CODD
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n -ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, join, retrieval language, predicate calculus, security, data integrity
CR CATEGORIES: 3.70, 3.73, 3.75, 4.20, 4.22, 4.29

1. Relational Model and Normal Form

1.1. INTRODUCTION

This paper is concerned with the application of elementary relation theory to systems which provide shared access to large banks of formatted data. Except for a paper by Childs [1], the principal application of relations to data systems has been to deductive question-answering systems. Levin and Maron [2] provide numerous references to work in this area.

In contrast, the problems treated here are those of *data independence*—the independence of application programs and terminal activities from growth in data types and changes in data representation—and certain kinds of *data inconsistency* which are expected to become troublesome even in nondeductive systems.

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for non-inferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the “connection trap”).

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

1.2. DATA DEPENDENCIES IN PRESENT SYSTEMS

The provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence [5, 6, 7]. Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed *without logically impairing some application programs* is still quite limited. Further, the model of data with which users interact is still cluttered with representational properties, particularly in regard to the representation of collections of data (as opposed to individual items). Three of the principal kinds of data dependencies which still need to be removed are: ordering dependence, indexing dependence, and access path dependence. In some systems these dependencies are not clearly separable from one another.

1.2.1. *Ordering Dependence.* Elements of data in a data bank may be stored in a variety of ways, some involving no concern for ordering, some permitting each element to participate in one ordering only, others permitting each element to participate in several orderings. Let us consider those existing systems which either require or permit data elements to be stored in at least one total ordering which is closely associated with the hardware-determined ordering of addresses. For example, the records of a file concerning parts might be stored in ascending order by part serial number. Such systems normally permit application programs to assume that the order of presentation of records from such a file is identical to (or is a subordering of) the

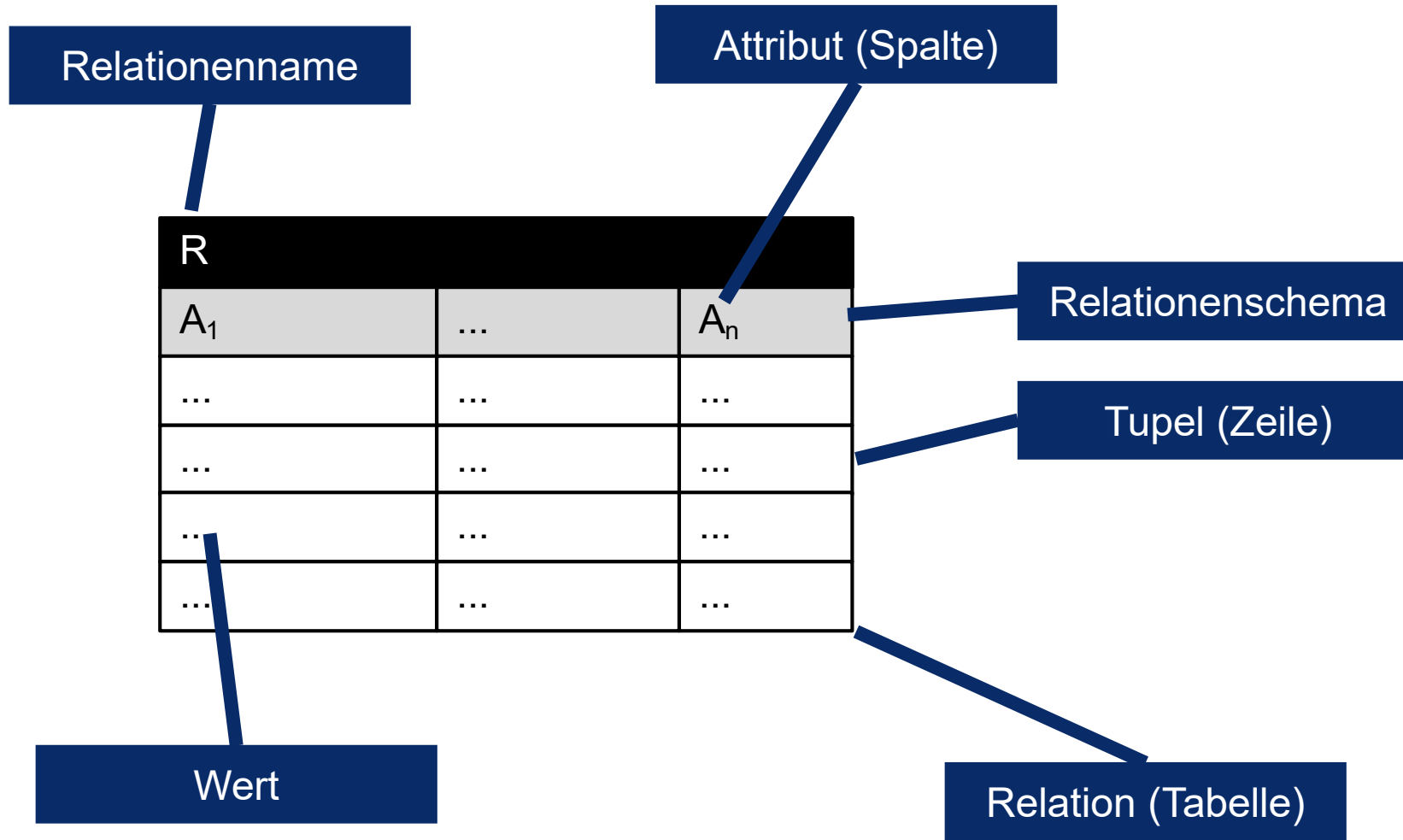
Volume 13 / Number 6 / June, 1970

Communications of the ACM 377

Codd (1970)

- 1** Das relationale Modell
- 2** Integritätsbedingungen
- 3** Vom ER-Modell zum relationalen Modell

Relationales Modell



- Seien A_1, A_2, \dots, A_n Attribute mit zugehörigen Domänen D_1, D_2, \dots, D_n
- Eine Relation ergibt sich aus dem kartesischen Produkt, d.h. $R \subseteq A_1 \times \dots \times A_n$
- Tupel $t = \langle v_1, v_2, \dots, v_n \rangle \in R$
- Wert $v \in \text{dom}(A_n) \mid \text{NULL}$

Unser Beispiel

- Angestellte AKW Springfield \subseteq Vorname x Nachname x ID
- $t = (\text{„Homer“}, \text{„Simpson“}, 3226)$
- Zustand $r = \{t_1, t_2, \dots, t_n\}$

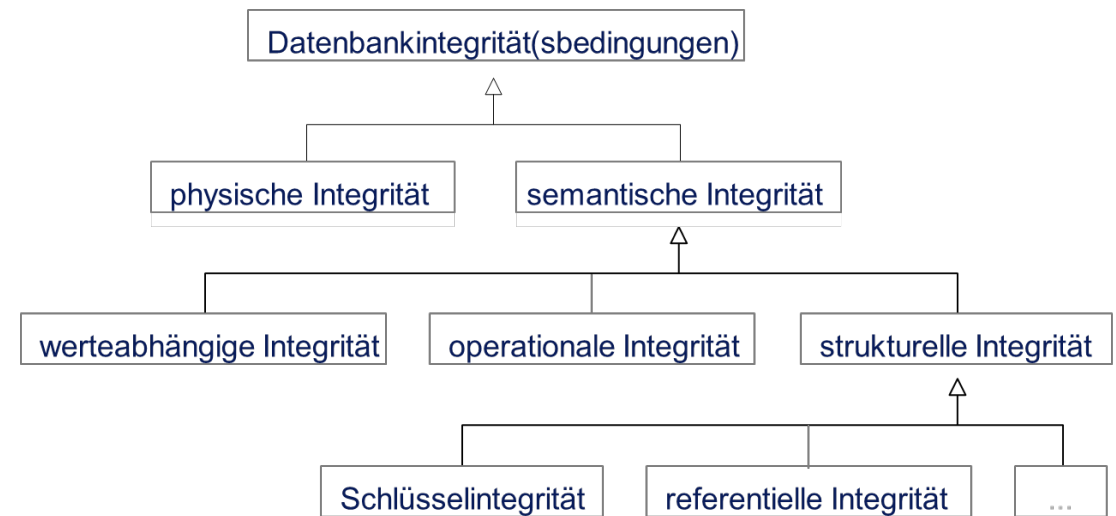
- **Schema:** Angestellte AKW Springfield (Vorname, Nachname, ID)

Angestellte AKW Springfield Vorname Nachname ID

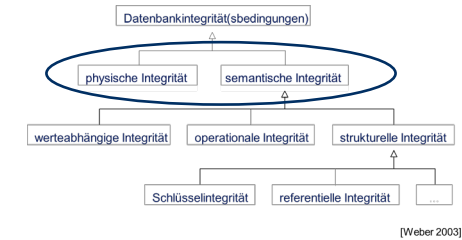
R		
A ₁	...	A _n
Homer	Simpson	3226
...
...
...

- 1 Das relationale Modell
- 2 Integritätsbedingungen
 - 2.1 Statische und dynamische Integrität
 - 2.2 Strukturelle Integrität
- 3 Vom ER-Modell zum relationalen Modell

- Die Integrität einer Datenbank d bezeichnet die korrekte und widerspruchsfreie Speicherung von Daten in d .
- Zur Einhaltung bzw. Gewährleistung der Integrität werden Integritätsbedingungen formuliert.
 - Die „Reichweite“ einer Integritätsbedingung, also die Anzahl der betroffenen Relationen, sollte möglichst gering gehalten werden.
 - Art und Umfang der realisierten Integritätskonzepte variieren deutlich in kommerziellen Datenbankprodukten.
 - Ein flexibler Mechanismus zur Sicherung komplexer Integritätsbedingungen sind Trigger. Ein Trigger ist eine benutzerdefinierte Funktion, die automatisch aufgerufen wird, sobald eine bestimmte Bedingung erfüllt ist. Einsatz für Trigger sind z.B. Tabellen mit Statistiken, in denen Werte abgeleiteter Attribute bzw. Spalten zu berechnen sind.



[Weber 2003]



Physische Integrität

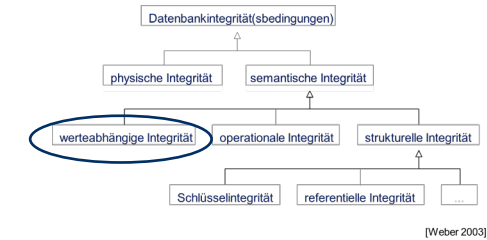
- Regelt alle Aspekte der physikalischen Datenspeicherung
- Berücksichtigt Hard- und Softwarefehler (soweit möglich)
- Behandelt Probleme des gleichzeitigen Zugriffs auf Daten.

Semantische Integrität

- Wird aus den Eigenschaften des zu modellierenden Weltausschnitts abgeleitet und definiert die zulässigen Zustände der Datenbank:
 - Legt fest, welche Werte erlaubt und welche Beziehungen zwischen Datenelementen möglich sind.
- Das Datenbank-Management-System überprüft mit semantischen Integritätsbedingungen, ob eine gewünschte Änderung der Datenbank zulässig ist.
- Semantische Integritätsbedingungen werden nach dem Inhalt unterschieden, den sie behandeln – in wertabhängige, operationale und strukturelle Integritätsbedingungen.

- 1** Das relationale Modell
- 2** Integritätsbedingungen
 - 2.1** Statische und dynamische Integrität
 - 2.2** Strukturelle Integrität
- 3** Vom ER-Modell zum relationalen Modell

Werteabhängige (= statische) Integrität



- Werteabhängige Integrität bezeichnet die Forderung, nicht alle durch Entity-Typen und Beziehungs-Typen definierten Datenbankzustände zuzulassen:
 - Werteabhängige Integritätsbedingungen treffen Aussagen über den Zusammenhang von Werten zwischen Datenelementen.
 - Sie beschränken die in der Anwendungsoberfläche evtl. möglichen Datenelemente auf Teilmengen.

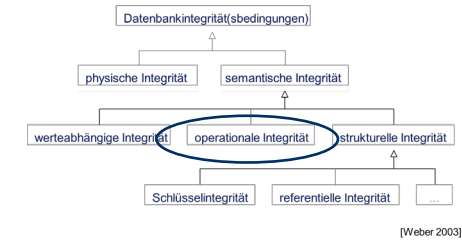
- Beispiele:
- Lokal, innerhalb eines Tupels oder einer Relation: Die PLZ muss zwischen 1000 und 99999 liegen.
- Global, zwischen Relationen: die Summe der Gehälter aller Mitarbeiter an einem Standort darf dessen Personalbudget nicht überschreiten.

Personal			
Name	PersNr.	Standort	Gehalt
Maier	12	Neuss	66.000
Steffen	14	Neuss	56.000
Pearl	24	Marl	67.000
...			

} "Σ<"

Produktion			
Standort	PLZ	Mitarbeiter	Budget (Mio)
Neuss	47323	111	44.3
Marl	45214	65	34.3
...			

Operationale (= dynamische) Integrität



Operationale Integritätsbedingungen gewährleisten korrekte Zustandsübergänge:

- Sie überwachen die Ausführung von Operationen.
- Durch die Angabe von Integritätsbedingungen wird die Ausführung einer Transaktion hinsichtlich der Einhaltung von Vor- und Nachbedingungen abhängig gemacht.
- Beispiele:
 - Vom Familienstand „ledig“ ist nur ein Übergang zu verheiratet möglich – und z.B. nicht zu „geschieden“.
 - Das Gehalt von Mitarbeitern darf nur steigen.

Zeitpunkt t:

Personal			
Name	PersNr.	Standort	Gehalt
Maier	12	Neuss	66.000
Steffen	14	Neuss	56.000
Pearl	24	Marl	67.000
...			

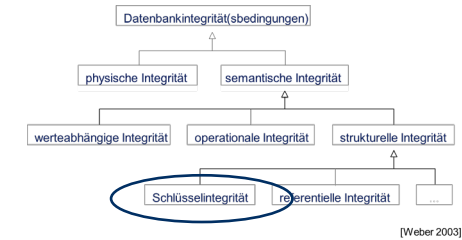
Zeitpunkt t+1

Personal			
Name	PersNr.	Standort	Gehalt
Maier	12	Neuss	67.000
Steffen	14	Neuss	59.000
Pearl	24	Marl	67.000
...			

"<"

- 1** Das relationale Modell
- 2** Integritätsbedingungen
 - 2.1** Statische und dynamische Integrität
 - 2.2** Strukturelle Integrität
- 3** Vom ER-Modell zum relationalen Modell

Schlüsselattribute (Wiederholung)



▪ Schlüssel

- **minimale** Menge von Attributen, deren Werte ein Tupel **immer eindeutig** identifizieren
- hat für jedes Tupel eine andere Ausprägung (UNIQUE)
- muss entsprechend existieren (NOT NULL)
- besondere Bedeutung bei der **Referenzierung** von Tupeln
- **einer** der Schlüsselkandidaten wird als Primärschlüssel ausgewählt

▪ Unterscheidung

- Merkmalschlüssel (z.B. Name)
- künstlicher Schlüssel (z.B. Bestellnummer)
- Kombination aus Schlüssel (z.B. Gebäude- und Raumnummer)

▪ Primärschlüssel

- wird **unterstrichen**
- Angestellte AKW Springfield (Vorname, Nachname, **ID**)

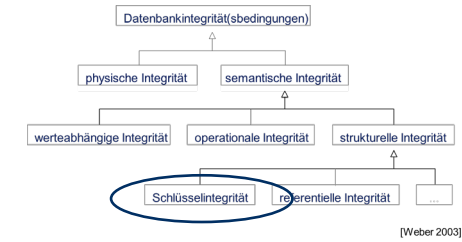
▪ Sekundärschlüssel

- auch Schlüssel**kandidaten** (candidate key) genannt
- sonst Eigenschaften wie Schlüssel

▪ Fremdschlüssel

- auch: relationale Verknüpfung
- wird mit **#** gekennzeichnet
- Attribut
 - (welches **nicht** alleiniger Primärschlüssel ist)
 - welches den selben Wertebereich hat wie ein Primärschlüssel in einer anderen Relation
 - welches beide Relationen miteinander verbindet

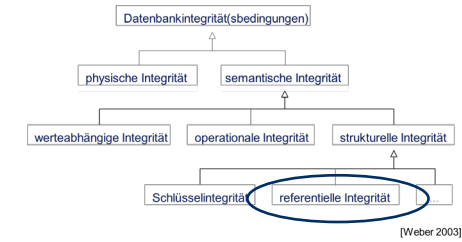
Beispiel Schlüssel und Schlüsselintegrität



- {Vorname, Nachname} und {PANr} sind Beispiele für Schlüsselkandidaten für die Relation „Personen“.
- Der Primärschlüssel wird durch Unterstreichen der entsprechenden Attribute gekennzeichnet.

Personen							
<u>PANr</u>	<u>Vorname</u>	<u>Nachname</u>	<u>PLZ</u>	<u>ORT</u>	<u>Straße</u>	<u>HNr.</u>	<u>Geb.Datum</u>
4711	Andreas	Heuer	18209	DBR	BHS	15	31.10.1958
5588	Gunter	Saake	39106	MD	STS	55	05.10.1960
6834	Michael	Korn	39104	MD	BS	41	24.09.1974
7754	Andreas	Möller	18209	DBR	RS	31	25.02.1976
8832	Tamara	Jagellovsk	38106	BS	GS	12	11.11.1973
9912	Antje	Hellhof	18059	HRO	AES	21	04.04.1970
9999	Christa	Loeser	69121	HD	TS	38	10.05.1969

Fremdschlüssel und referentielle Integrität



- Seien $r_1(R_1)$ und $r_2(R_2)$ zwei Relationen mit den Schemata R_1 bzw. R_2 .
- Sei weiterhin κ Primärschlüssel von R_1 .
- Dann ist $a \subset R_2$ Fremdschlüssel in r_2 bezüglich κ in r_1 , falls für alle Tupel $t_2 \in r_2$ gilt:

$$\exists t_1 \in r_1 \quad \text{mit} \quad t_1(\kappa) = t_2(a)$$

$$\text{bzw.} \quad \{ t_2(a) \mid t_2 \in r_2 \} \subseteq \{ t_1(\kappa) \mid t_1 \in r_1 \}$$

Die Erfüllung dieser Eigenschaft heißt referentielle Integrität, Fremdschlüsselintegrität oder auch Fremdschlüsselbedingung.

$$R_2 = \{ B_1, B_2, B_3, \dots, B_m \}$$

Fremd "schlüssel" α

B ₁	B ₂	B ₃	...	B _m

$r_2(R_2)$

$$R_1 = \{ A_1, A_2, A_3, \dots, A_n \}$$

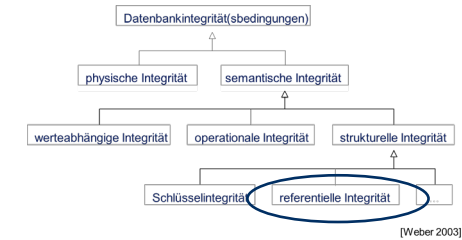
Schlüssel κ

A ₁	A ₂	A ₃	...	A _n

$r_1(R_1)$

$$\{ t_2(\alpha) \mid t_2 \in r_2 \} \subseteq \{ t_1(\kappa) \mid t_1 \in r_1 \}$$

Fremdschlüssel und referentielle Integrität (Beispiel)

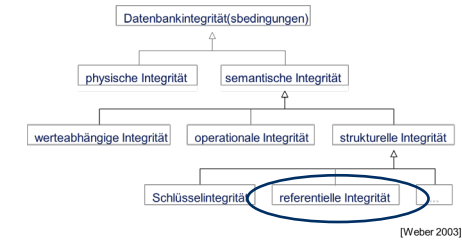


Personen							
PANr	Vorname	Nachname	PLZ	Ort	Straße	HNr.	Geb.Datum
4711	Andreas	Heuer	18209	DBR	BHS	15	31.10.1958
5588	Gunter	Saake	39106	MD	STS	55	05.10.1960
6834	Michael	Korn	39104	MD	BS	41	24.09.1974
7754	Andreas	Möller	18209	DBR	RS	31	25.02.1976
8832	Tamara	Jagellovsk	38106	BS	GS	12	11.11.1973
9912	Antje	Hellhof	18059	HRO	AES	21	04.04.1970
9999	Christa	Loeser	69121	HD	TS	38	10.05.1969

Pers_Telefon	
PANr	Telefon
4711	038203-12230
4711	0381-498-3401
4711	0381-498-3427
5588	0391-34677
5588	0391-5592-3800
9999	06221-400177

- {PANr} ist Schlüssel für die Relation „Personen“, {PANr, Telefon} ist Schlüssel für die Relation „Pers_Telefon“
- {PANr} in der Relation „Pers_Telefon“ ist ein Fremdschlüssel und nimmt Bezug auf {PANr} in der Relation „Personen“.
- Fremdschlüsselbedingung: $\forall t_2 \in \text{Pers_Telefon} \text{ gilt: } \exists t_1 \in \text{Personen} \text{ mit } t_1(\text{PANr}) = t_2(\text{PANr})$ Iternativ: $\{t_2(\text{PANr}) \mid t_2 \in \text{Pers_Telefon}\} \subseteq \{t_1(\text{PANr}) \mid t_1 \in \text{Personen}\}$

Fremdschlüssel und referentielle Integrität (DB-Verhalten)



Sei $r_1(R_1)$ eine Relation mit Primärschlüssel κ und sei $r_2(R_2)$ eine Relation mit Fremdschlüssel a bezüglich κ in r_1 .

Gewährleistung der referentiellen Integrität erfordert:

1. Das Einfügen eines Tupels in r_2 verlangt, dass der Fremdschlüssel auf ein existierendes Tupel in r_1 verweist.
 - Das automatische Anlegen eines Tupels in r_1 mit der entsprechenden Fremdschlüsselausprägung wird *Cascading Insert* genannt.
2. Das Ändern eines Tupels in r_2 verlangt, dass die neue Fremdschlüsselausprägung auf ein existierendes Tupel in r_1 verweist.

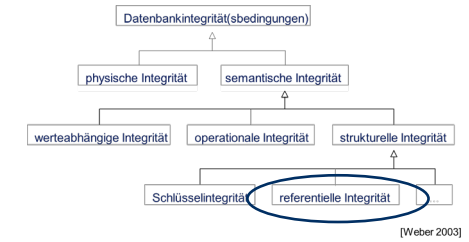
3. Das Ändern der Primärschlüsselausprägung für ein Tupel in r_1 verlangt, dass kein Tupel aus r_2 auf dieses verwiesen hat.
 - Das automatische Nachvollziehen solcher Änderungen wird *Cascading Update* genannt.
4. Das Löschen eines Tupels in r_1 verlangt, dass kein Tupel aus r_2 auf dieses verwiesen hat.
 - Das Verhindern der Löschung von Tupeln aus r_1 solange noch hierauf bezugnehmende Tupel in r_2 vorhanden sind, wird *Prohibited Delete* genannt. Die Propagierung von Löschungen in r_1 weiter zu r_2 wird als *Cascading Delete* bezeichnet.

(Fehlende) referentielle Integrität in der Praxis



Aber

- Lediglich client-side Check
- Name wurde als Text kopiert
- Es wurde **keine** Fremdschlüsselzuweisung gemacht



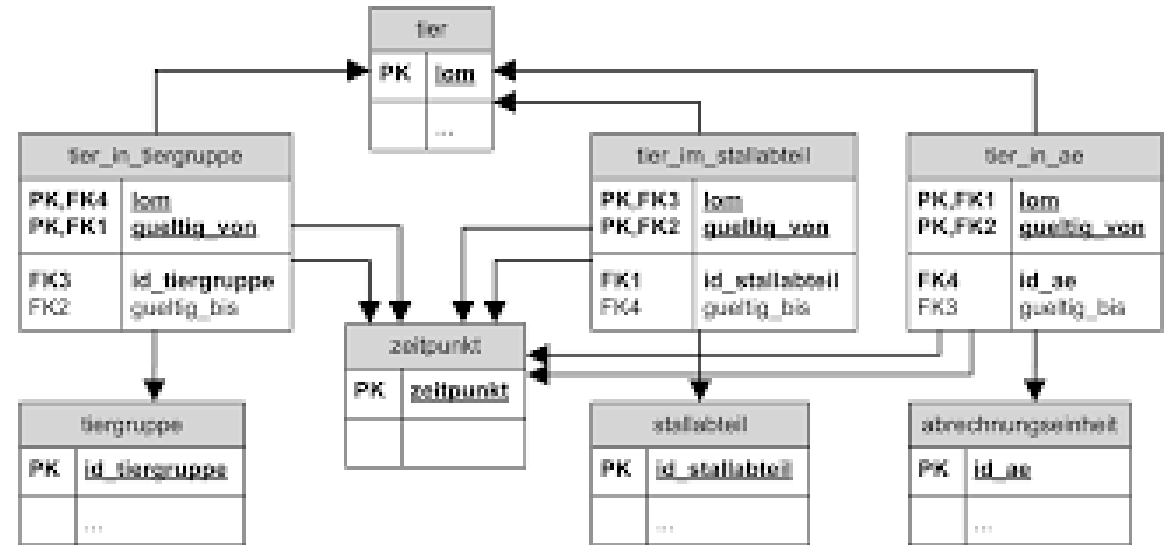
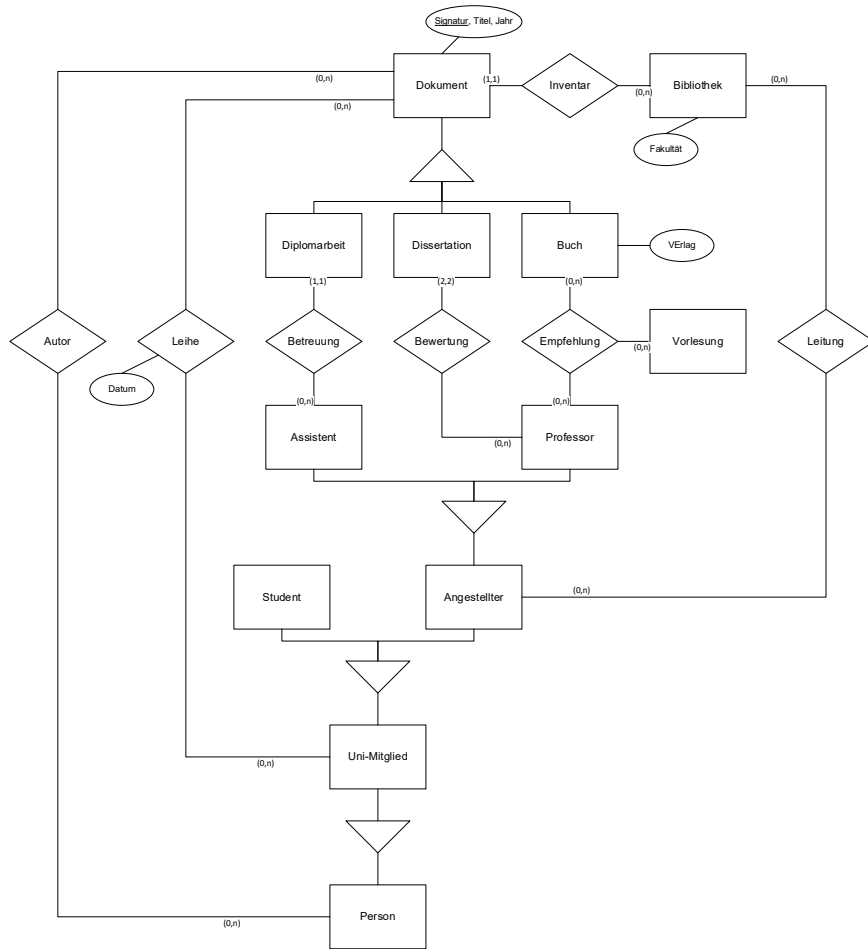
Idee

- Individueller Aufdruck auf Flasche über meinecoke.de
- Whitelist mit 30.000 Namen



- 1 Das relationale Modell
- 2 Integritätsbedingungen
- 3 Vom ER-Modell zum relationalen Modell
 - 3.1 Transformation von Entitäten
 - 3.2 Transformation von Beziehungen

ERM fertig und nun?



- Das ER-Modell besitzt zwei grundlegende Strukturierungskonzepte:
 1. Entity-Typen $E(A_1, \dots, A_n)$
 2. Beziehungstypen $R(E_1, \dots, E_m; A_1, \dots, A_n)$
- Im relationalen Modell werden beide auf das einzige Strukturierungskonzept „Relationenschema“, R , abgebildet.
 - Dabei dient das Konzept der Fremdschlüssel zur Abbildung von Beziehungstypen.



- 1** Das relationale Modell
- 2** Integritätsbedingungen
- 3** Vom ER-Modell zum relationalen Modell
 - 3.1** Transformation von Entitäten
 - 3.2** Transformation von Beziehungen

Transformation Entität (allgemein)



Umsetzung:

1. Dem Entity-Typ E wird Relationenschema R_E zugeordnet.
Die Attribute A_1, \dots, A_n von E werden Attribute von R_E .
2. Der Schlüssel $\kappa \subseteq \{A_1, \dots, A_n\}$ von E wird Primärschlüssel von R_E .

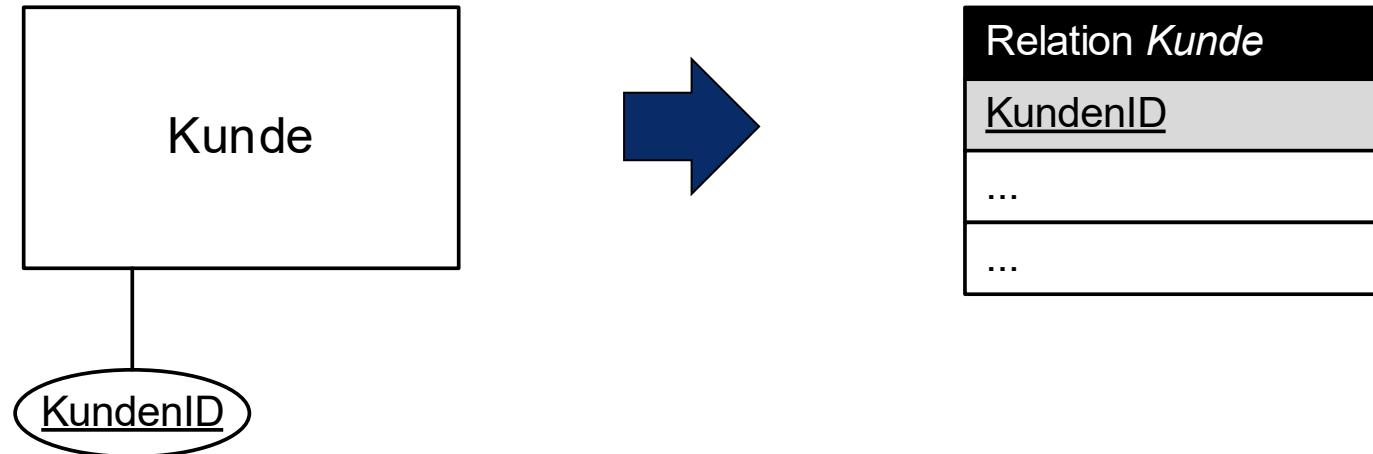
Alternative:

Festlegen eines formalen Primärschlüssels durch Hinzufügen eines Schlüsselattributes ID zur Umsetzung der Eindeutigkeit von Entitäten. Der ursprüngliche Schlüssel κ ist dann ein weiterer Schlüssel im Relationenschema R_E .

3. Der Primärschlüssel wird durch Unterstreichen gekennzeichnet.

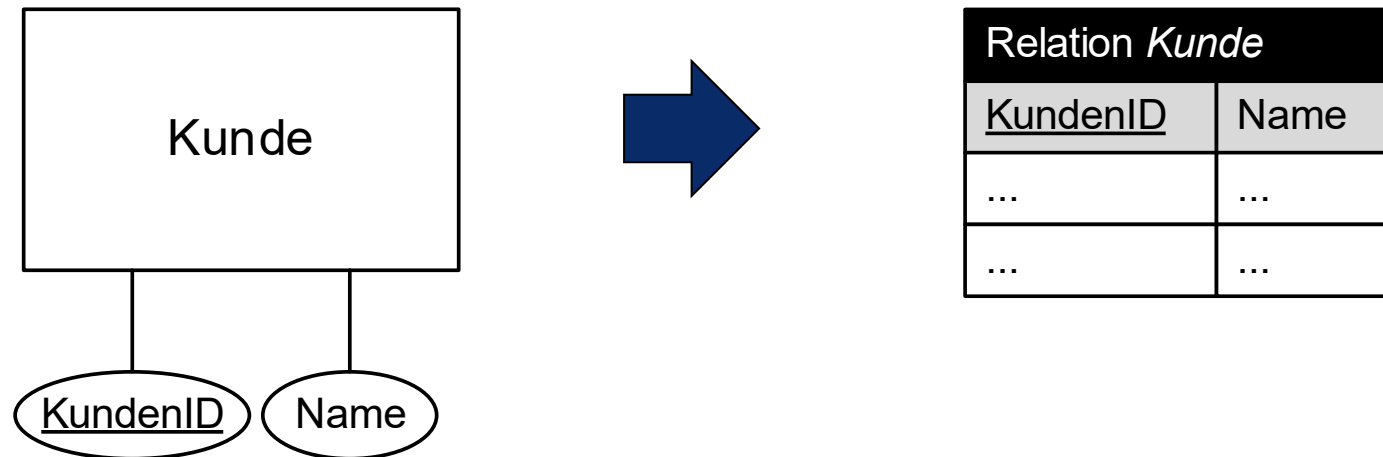
Transformation Entität (Beispiel)

- **Entitäten** werden immer zu eigenen Relationen
 - Tabellen mit mindestens einer Spalte
 - Spalte des Primärschlüssels



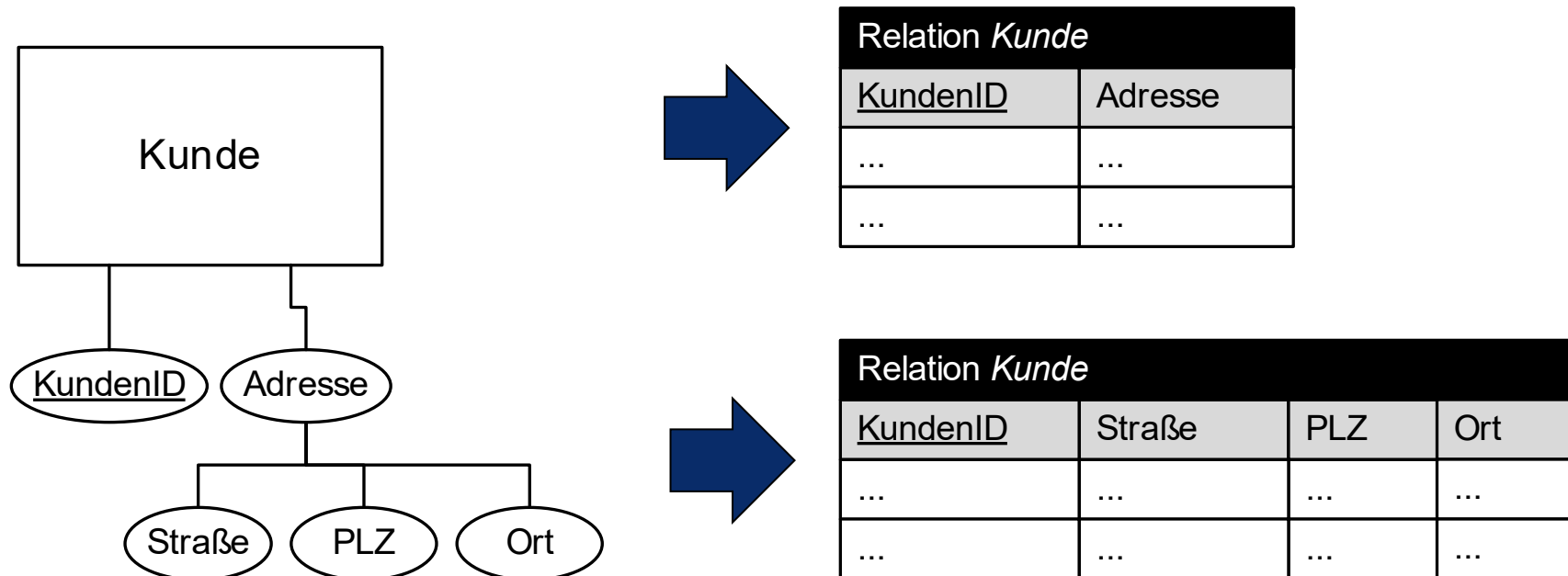
Transformation Attribut (Beispiel)

- **Attribute** werden immer zu Spalten in Relationen
 - Weiteren Spalten neben dem Primärschlüssel



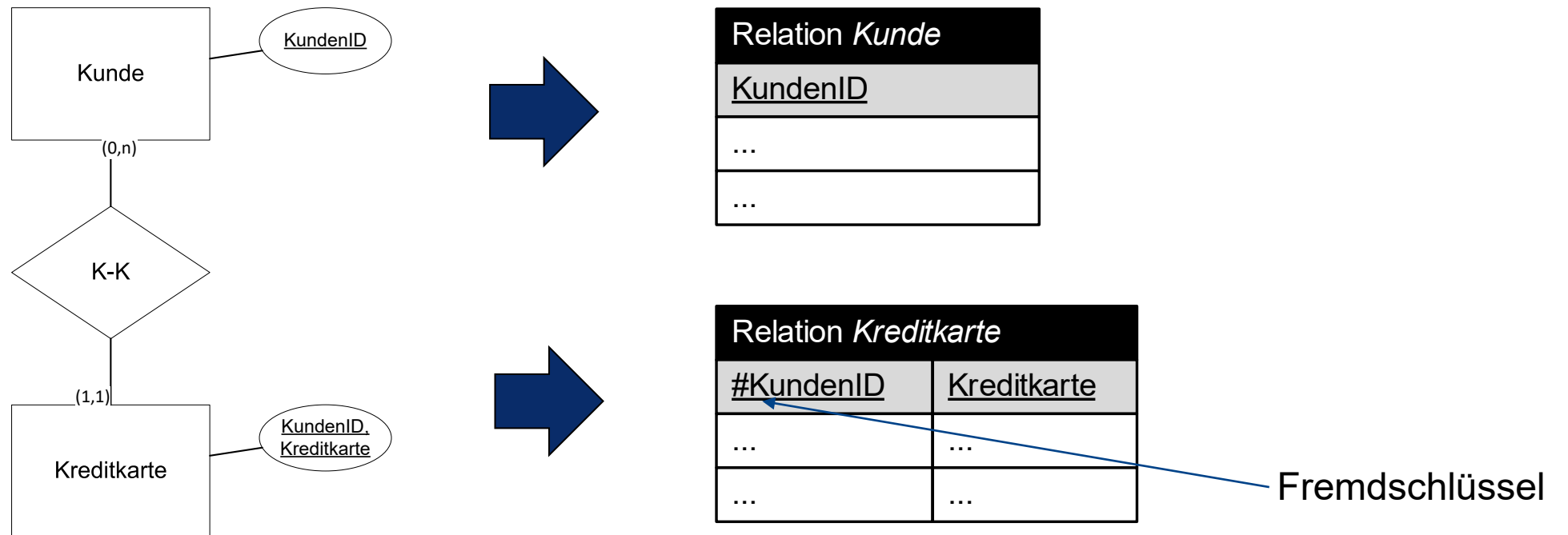
Transformation zusammengesetztes Attribut (Beispiel)

- **Zusammengesetzte Attribute** werden zu Spalten in Relationen, indem man
 - die Komponenten eliminiert oder
 - die Komponenten flachklopft



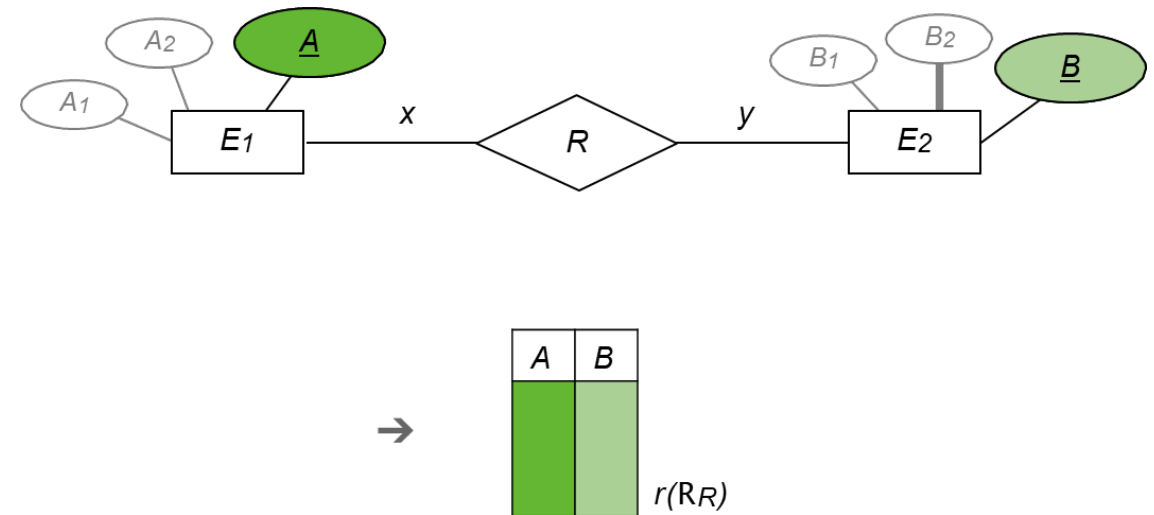
Transformation mehrwertiges Attribut (Beispiel)

- **Mehrwertige Attribute** werden zu neuen Relationen mit Trivialschlüssel
 - Jedes mehrwertige Attribut wird eine Relation
 - Primärschlüssel wird der Schlüssel der ursprünglichen Relation sowie das Attribut selbst

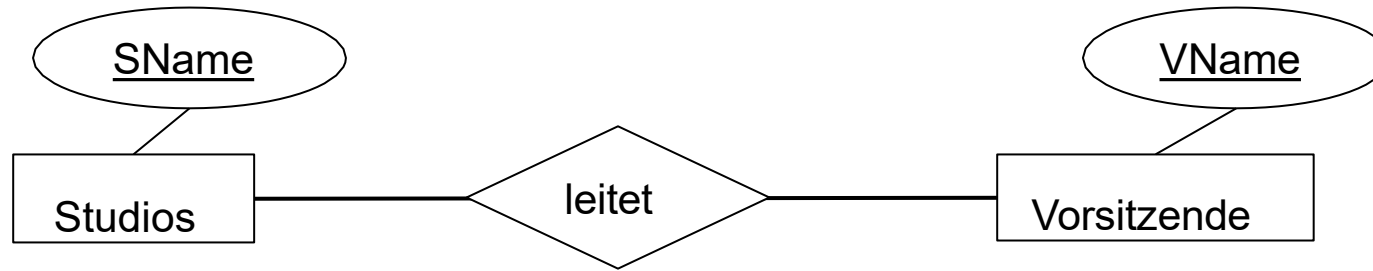


- 1 Das relationale Modell
- 2 Integritätsbedingungen
- 3 Vom ER-Modell zum relationalen Modell
 - 3.1 Transformation von Entitäten
 - 3.2 Transformation von Beziehungen
 - a $(*,*)-(x,n)$
 - b $(*,1)-(*,1)$
 - c **Generalisierung / Spezialisierung**

- Exakte Darstellung der Informationen des ER-Modells
 - Das Datenbankschema kann genauso viele Instanzen wie das ER-Diagramm darstellen.
 - Das Datenbankschema kann nicht mehr Instanzen als das ER-Diagramm darstellen.
- Die dahinter stehende Forderung bei der Abbildung von Beziehungstypen ist die **Kapazitätserhaltung**:
 - Alle Instanzen des ER-Modells sind auch Instanzen des relationalen Modells und umgekehrt.
 - Gibt es eine bijektive Abbildung zwischen den Instanzen eines Entity-Relationship-Modells und den Instanzen eines relationalen Modells, so nennt man die Transformation zwischen den Modellen kapazitätserhaltend.
- Transformationsregeln beschreiben die Überführung von ERM-Konstrukten in Relationen
 - Aber Transformationsregeln sind **nicht immer eindeutig!**



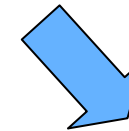
Kapazitätserhöhende Abbildung



Relationenschema: $R = \{SName, VName\}$



Schlüsselmenge: $\{ \{SName\} \}$



Schlüsselmenge: $\{ \{SName\}, \{VName\} \}$

SName	VName
Fox	Iger
Disney	Iger

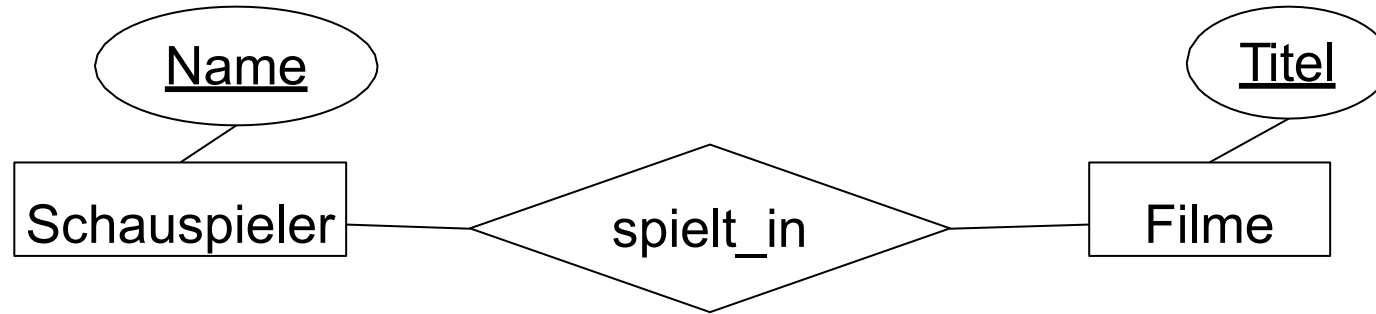
kapazitätserhöhend

SName	VName
Fox	Murdoch
Disney	Iger

SName	VName
Fox	Murdoch
Disney	Iger

kapazitätserhaltend

Kapazitätsvermindernde Abbildung



Relationenschema: $R = \{ \text{Name}, \text{Titel} \}$

Schlüsselmenge: $\{ \{ \text{Name} \} \}$

Name	Titel
Sharon Stone	Basic Instinct
Michael Douglas	Basic Instinct

kapazitätsvermindernd

Schlüsselmenge: $\{ \{ \text{Name}, \text{Titel} \} \}$

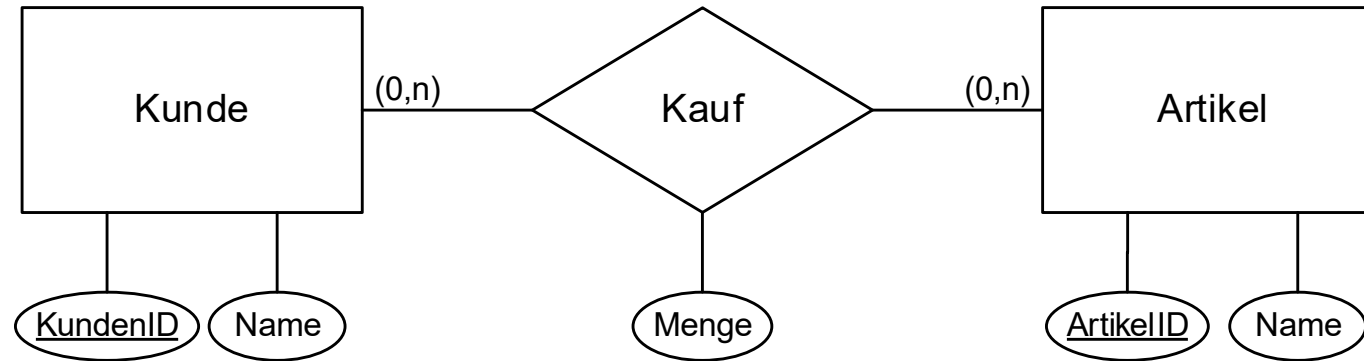
Name	Titel
Sharon Stone	Basic Instinct
Michael Douglas	Basic Instinct

kapazitäts-
erhaltend

Name	Titel
Sharon Stone	Basic Instinct
Sharon Stone	Total Recall
Michael Douglas	Basic Instinct

- 1 Das relationale Modell
- 2 Integritätsbedingungen
- 3 Vom ER-Modell zum relationalen Modell
 - 3.1 Transformation von Entitäten
 - 3.2 Transformation von Beziehungen
 - a $(*,*)-(x,n)$
 - b $(*,1)-(*,1)$
 - c **Generalisierung / Spezialisierung**

(x,n)-(x,n)-Zuordnung ($x \in \{0,1\}$)



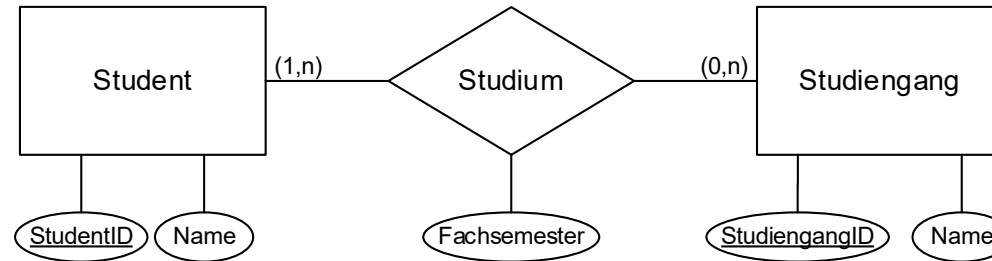
! Ohne künstlichen Schlüssel (Bestellnummer) kann jeder Artikel nur ein mal gekauft werden!

Relation <i>Kunde</i>	
<u>KundenID</u>	Name
...	...
...	...

Relation <i>Kauf</i>		
<u>#KundenID</u>	<u>#ArtikelID</u>	Menge
...
...

Relation <i>Artikel</i>	
<u>ArtikelID</u>	Name
...	...
...	...

Zwangseinträge (1)

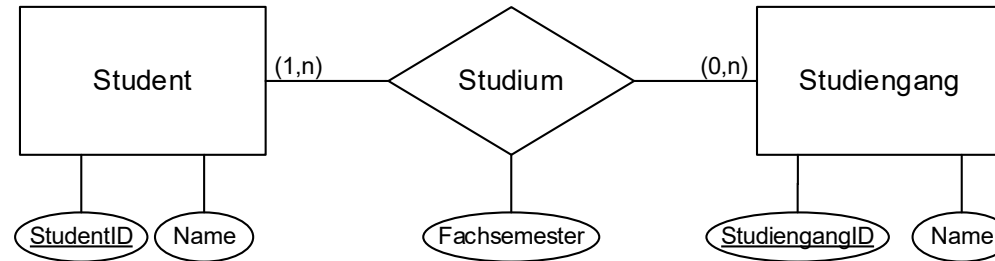


Relation <i>Student</i>	
<u>StudentID</u>	Name
1	Ronny
2	Silvio
3	Rocco
...	...

Relation <i>Studium</i>		
<u>#StudentID</u>	<u>#StudiengangID</u>	Fachsemester
1	1	2
2	3	16
3	2	5
3	3	13
...

Relation <i>Artikel</i>	
<u>StudiengangID</u>	Name
1	Kunstgeschichte
2	Hauswirtschaftslehre
3	Ökotrphologie
4	Wirtschaftsinformatik
...	...

Zwangseinträge (2)

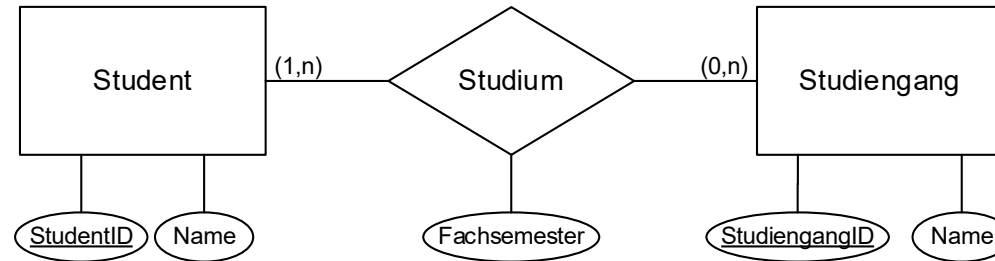


Relation <i>Student</i>	
<u>StudentID</u>	Name
1	Ronny
2	Silvio
3	Rocco
4	Kevin
...	...

Relation <i>Studium</i>		
<u>#StudentID</u>	<u>#StudiengangID</u>	Fachsemester
1	1	2
2	3	16
3	2	5
3	3	13
...

Relation <i>Artikel</i>	
<u>StudiengangID</u>	Name
1	Kunstgeschichte
2	Hauswirtschaftslehre
3	Ökotropologie
4	Wirtschaftsinformatik
...	...

Zwangseinträge (3)

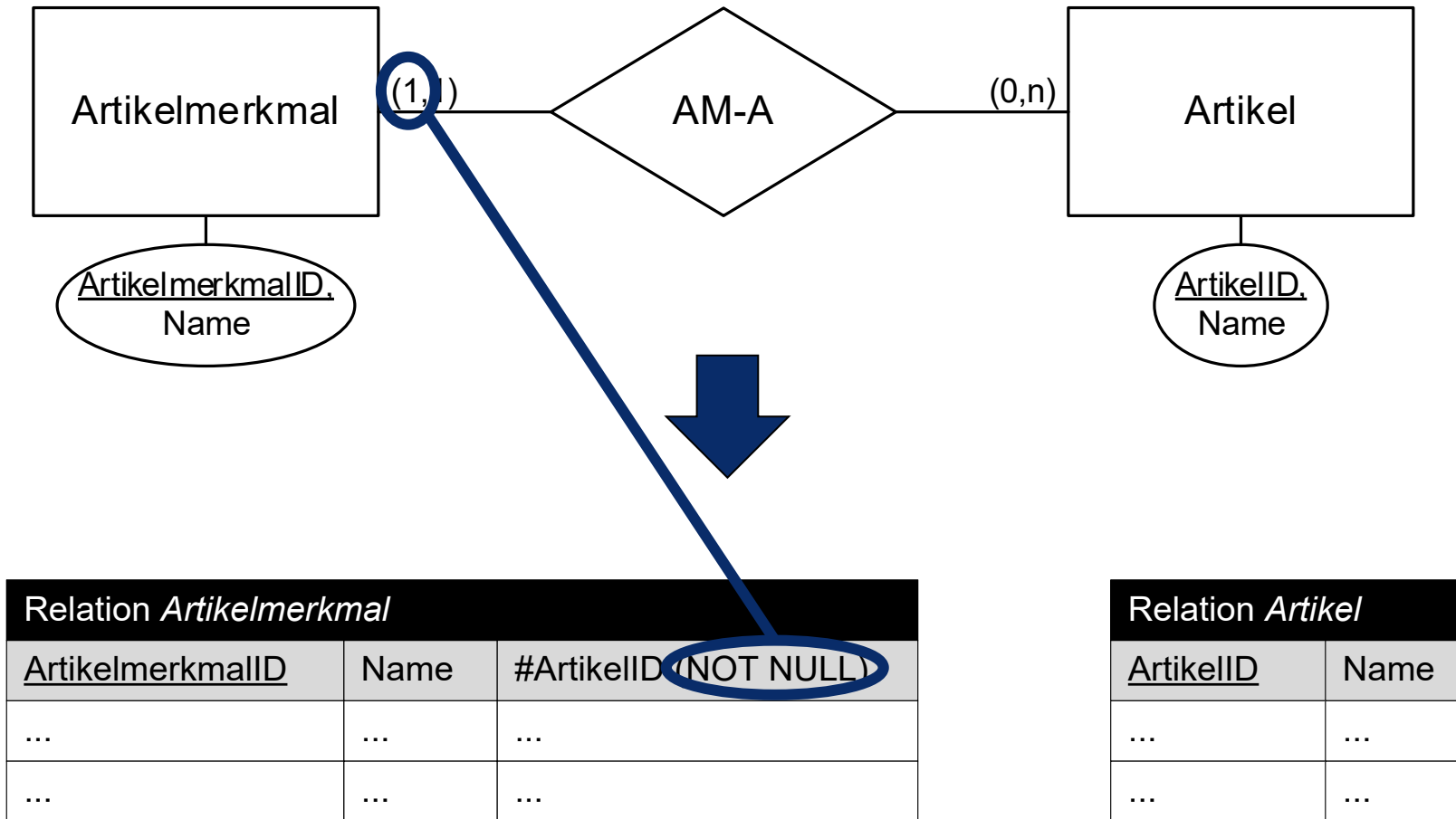


Relation <i>Student</i>	
<u>StudentID</u>	Name
1	Ronny
2	Silvio
3	Rocco
4	Kevin
...	...

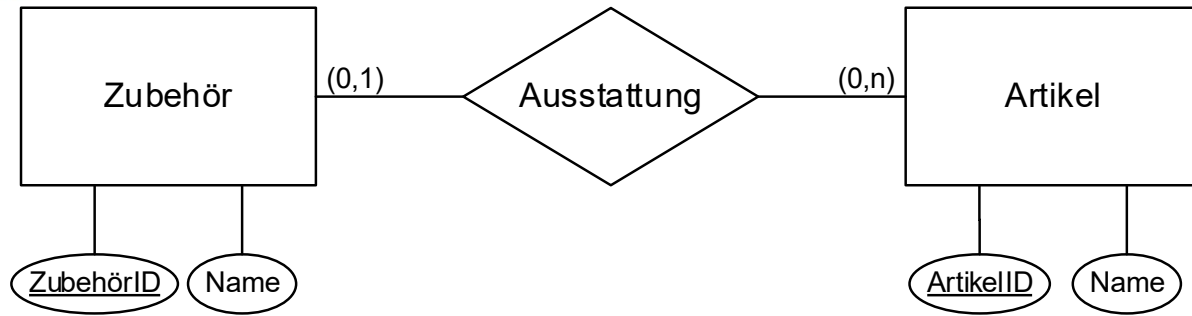
Relation <i>Studium</i>		
<u>#StudentID</u>	<u>#StudiengangID</u>	Fachsemester
1	1	2
2	3	16
3	2	5
3	3	13
4	3	1
...

Relation <i>Artikel</i>	
<u>StudiengangID</u>	Name
1	Kunstgeschichte
2	Hauswirtschaftslehre
3	Ökotropologie
4	Wirtschaftsinformatik
...	...

(1,1)-(x,n)-Zuordnung ($x \in \{0,1\}$)



(0,1)-(x,n)-Zuordnung ($x \in \{0,1\}$)



Relation <i>Zubehör</i>	
<u>ZubehörID</u>	Name
1	Spikereifen
2	Rennlenkeraufsatz
3	Fahrradhelm
...	...

Relation <i>Ausstattung</i>	
<u>#ZubehörID</u>	#ArtikelID
1	1
2	2
...	...

Relation <i>Artikel</i>	
<u>ArtikelID</u>	Name
1	Mountainbike
2	Rennrad
...	...

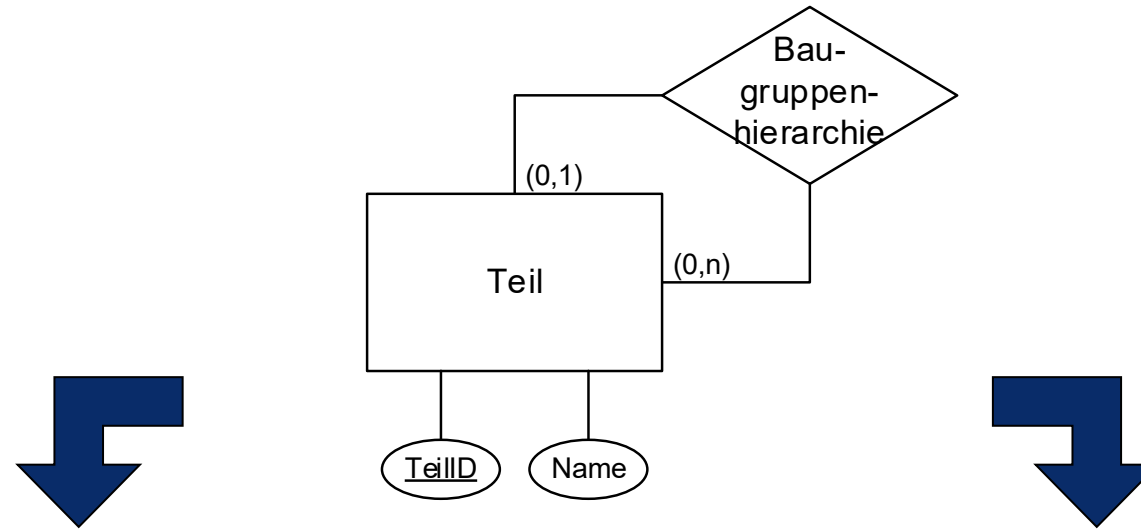
Relation <i>Zubehör</i>		
<u>ZubehörID</u>	Name	#ArtikelID
1	Spikereifen	1
2	Rennlenkeraufsatz	2
3	Fahrradhelm	NULL
...

Relation <i>Artikel</i>	
<u>ArtikelID</u>	Name
1	Mountainbike
2	Rennrad
...	...

1. Möglichkeit

2. Möglichkeit

(0,1)-(x,n)-Zuordnung als Hierarchie ($x \in \{0,1\}$)



1. Möglichkeit

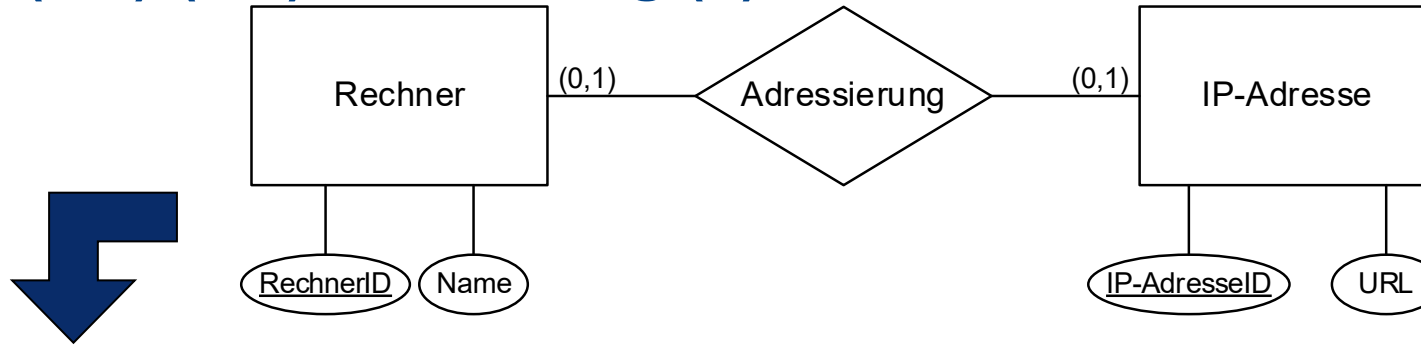
Relation <i>Teil</i>		
<u>TeilID</u>	Name	ÜG#TeilID
1	Fahrrad	NULL
2	Rad	1
3	Rahmen	1
4	Felge	2
...

2. Möglichkeit

Relation <i>Teil</i>		Relation <i>Baugruppenhierarchie</i>	
<u>TeilID</u>	Name	#TeilID	ÜG#TeilID
1	Fahrrad	2	1
2	Rad	3	1
3	Rahmen	4	2
4	Felge
...

- 1 Das relationale Modell
- 2 Integritätsbedingungen
- 3 Vom ER-Modell zum relationalen Modell
 - 3.1 Transformation von Entitäten
 - 3.2 Transformation von Beziehungen
 - a $(*,*)-(x,n)$
 - b $(*,1)-(*,1)$
 - c **Generalisierung / Spezialisierung**

(0,1)-(0,1)-Zuordnung (1)



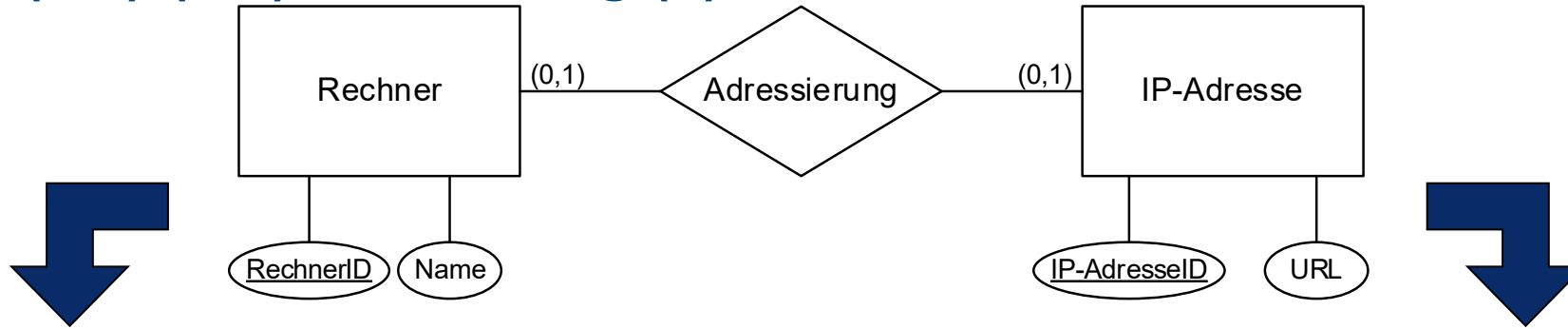
1. Möglichkeit

Relation <i>Rechner</i>		Relation <i>IP-Adresse</i>		
<u>RechnerID</u>	Name	<u>IP-AdresseID</u>	URL	#RechnerID (UNIQUE)
1	WWIN033	132.187.102.33	wwin033.wiinf.uni-wuerzburg.de	1
2	WI-BWLJP1	132.187.1.114	www.uni-wue.de	3
3	WRZ1114	80.237.133.158	www.wirtschaftsinformatik.de	NULL
...

2. Möglichkeit

Relation <i>Rechner</i>			Relation <i>IP-Adresse</i>	
<u>RechnerID</u>	Name	#IP-AdresseID (UNIQUE)	<u>IP-AdresseID</u>	URL
1	WWIN033	132.187.102.33	132.187.1.114	www.uni-wue.de
2	WI-BWLJP1	NULL	132.187.102.33	wwin033.wiinf.uni-wuerzburg.de
3	WRZ1114	132.187.1.114	128.176.0.12	www.wirtschaftsinformatik.de
...

(0,1)-(0,1)-Zuordnung (2)



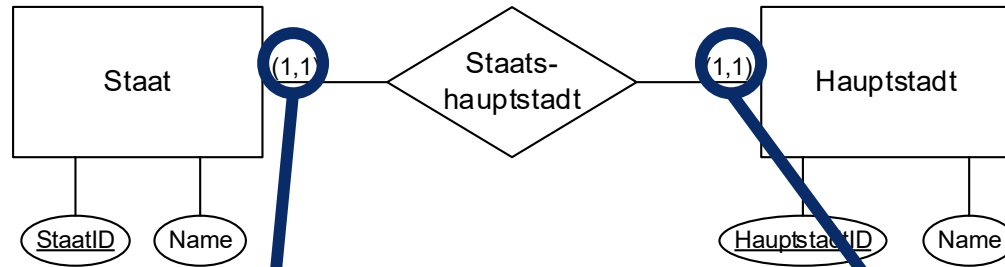
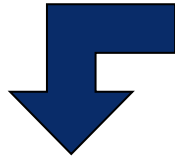
3. Möglichkeit

Relation <i>Rechner</i>		Relation <i>Adressierung</i>		Relation <i>IP-Adresse</i>	
<u>RechnerID</u>	Name	<u>#RechnerID</u>	<u>#IP-AdresseID (UNIQUE)</u>	<u>IP-AdresseID</u>	URL
1	WWIN033	1	132.187.102.33	132.187.102.33	wwin033.wiinf.uni-wuerzburg.de
2	WI-BWLJP1	3	132.187.1.114	132.187.1.114	www.uni-wue.de
3	WRZ1114	80.237.133.158	www.wirtschaftsinformatik.de
...

4. Möglichkeit

Relation <i>Rechner</i>		Relation <i>Adressierung</i>		Relation <i>IP-Adresse</i>	
<u>RechnerID</u>	Name	<u>#IP-AdresseID</u>	<u>#RechnerID (UNIQUE)</u>	<u>IP-AdresseID</u>	URL
1	WWIN033	132.187.102.33	1	132.187.102.33	wwin033.wiinf.uni-wuerzburg.de
2	WI-BWLJP1	132.187.1.114	3	132.187.1.114	www.uni-wue.de
3	WRZ1114	80.237.133.158	www.wirtschaftsinformatik.de
...

(1,1)-(1,1)-Zuordnung (1)



Relation Staat		
StaatID	Name	#HauptstadtID (UNIQUE, NOT NULL)
1	Vanuatu	3
2	Malta	2
3	Liechtenstein	1
...

Relation Hauptstadt	
HauptstadtID	Name
1	Vaduz
2	Valletta
3	Port Vila
...	...

1. Möglichkeit

Achtung: Bei neuen Einträgen in die eine Relation ist ein Trigger notwendig, der das Anlegen eines zugehörigen Eintrags in der anderen Relation erzwingt.

Relation Staat	
StaatID	Name
1	Vanuatu
2	Malta
3	Liechtenstein
...	...

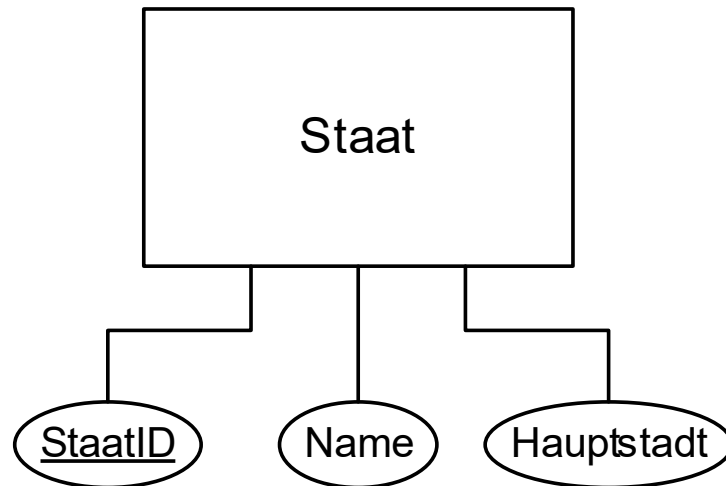
Relation Hauptstadt		
HauptstadtID	Name	#StaatID (UNIQUE, NOT NULL)
1	Vaduz	3
2	Valletta	2
3	Port Vila	1
...

2. Möglichkeit



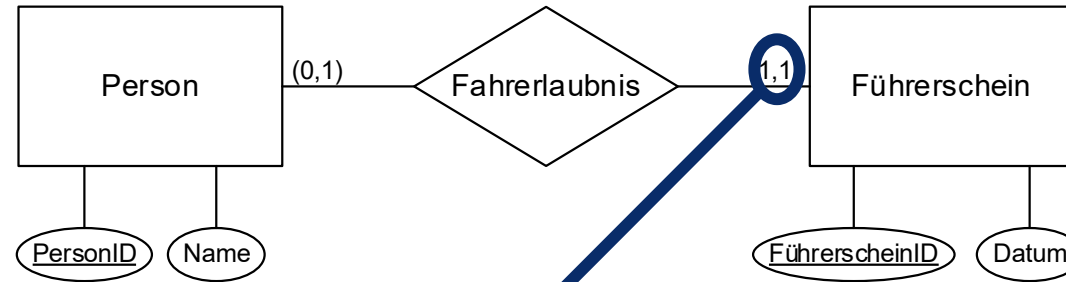
(1,1)-(1,1)-Zuordnung (2)

- (1,1)-(1,1)-Zuordnungen möglichst vermeiden
- Besser: Modellierung als Attribute:



Relation <i>Staat</i>		
<u>StaatID</u>	Name	Hauptstadt (NOT NULL)
1	Vanuatu	Port Vila
2	Malta	Valletta
3	Liechtenstein	Vaduz
...

(0,1)-(1,1)-Zuordnung



1. Möglichkeit

Relation Person	
PersonID	Name
1	Horst-Kevin
2	Tabea-Joyce
3	Klara Malvine
4	Savannah-Julienne
...	...

Relation Führerschein		
FührerscheinID	Datum	#PersonID (UNIQUE, NOT NULL)
123456789	01.01.2007	3
987654321	29.04.2010	1
111111111	10.12.2013	2
...

2. Möglichkeit

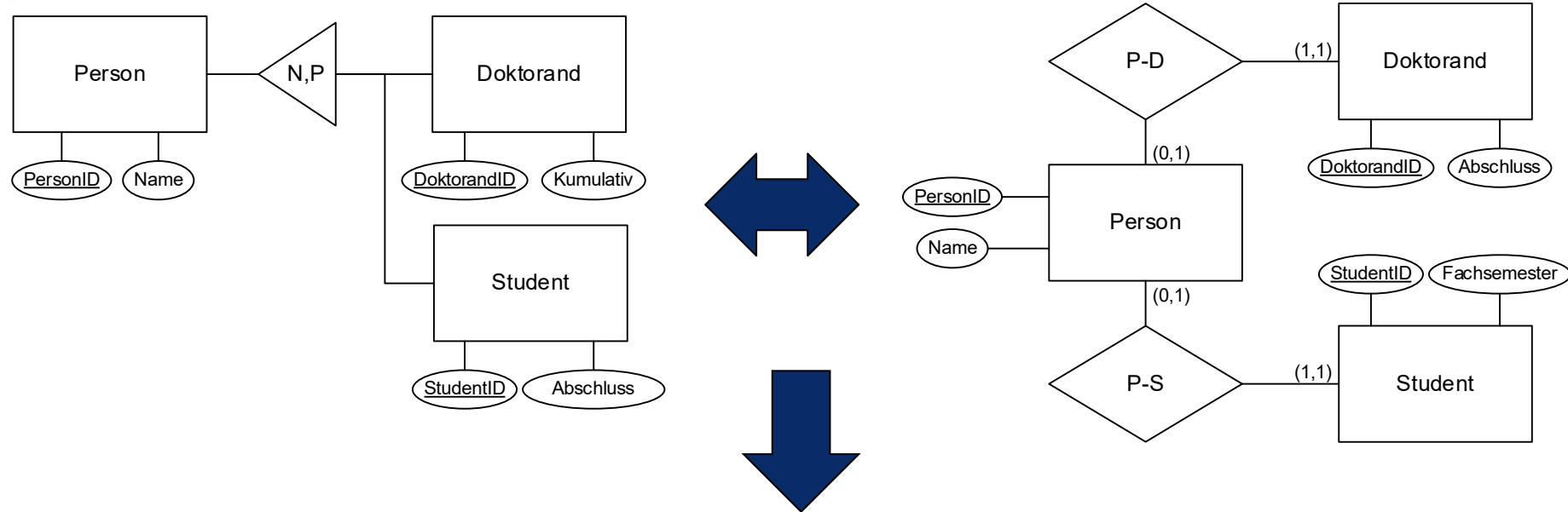
Relation Person		
PersonID	Name	#Führerschein.ID (UNIQUE)
1	Horst-Kevin	987654321
2	Tabea-Joyce	111111111
3	Klara Malvine	123456789
4	Savannah-Julienne	NULL
...

Relation Führerschein	
FührerscheinID	Datum
123456789	01.01.2007
987654321	29.04.2010
111111111	10.12.2013
...	...

Achtung: Bei neuen Einträgen in Führerschein ist ein Trigger notwendig. Sobald ein neuer Führerschein angelegt wird, muss er in der anderen Tabelle einer Person zugeordnet werden

- 1** Das relationale Modell
- 2** Integritätsbedingungen
- 3** Vom ER-Modell zum relationalen Modell
 - 3.1** Transformation von Entitäten
 - 3.2** Transformation von Beziehungen
 - a** $(*,*)-(x,n)$
 - b** $(*,1)-(*,1)$
 - c** **Generalisierung / Spezialisierung**

Generalisierung/ Spezialisierung (N,P)



Relation <i>Person</i>			
<u>PersonID</u>	Name	#DoktorandID (UNIQUE)	#StudentID (UNIQUE)
1	Ronny	D1	NULL
2	Silvio	NULL	S1
3	Rocco	D2	S3
4	Kevin	NULL	NULL
5	Mike	NULL	S2
...

Relation <i>Doktorand</i>	
<u>DoktorandID</u>	Abschluss
D1	Master
D2	Diplom
...	...

Relation <i>Student</i>	
<u>StudentID</u>	Fachsemester
S1	3
S2	6
S3	15
...	...

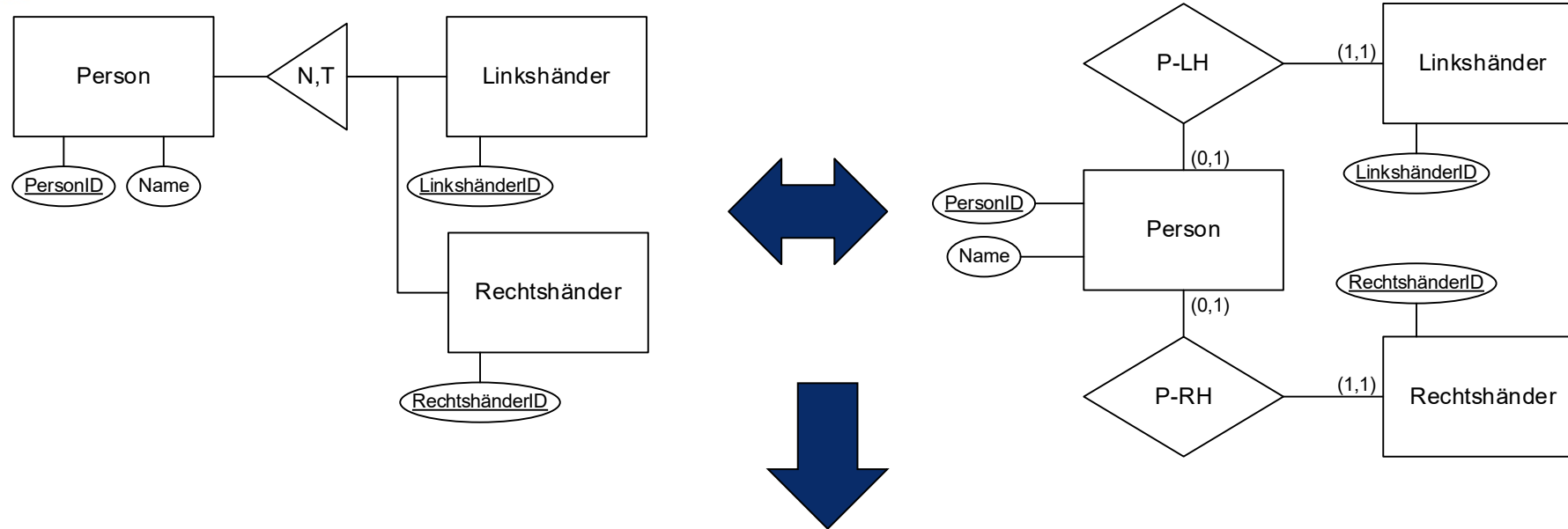
Nicht disjunkt-Partiell: Jede Entität *kann einem oder mehreren* Spezialfällen angehören!

Im folgenden werden stets Trigger benötigt

- **Achtung:** Trigger für Neueinträge in die spezialisierten Tabellen notwendig
 - vgl. (0,1)-(1,1)-Führerschein-Beispiel



Generalisierung/ Spezialisierung (N,T)



Relation <i>Person</i>			
<u>PersonID</u>	Name	#LinkshänderID (UNIQUE)	#RechtshänderID (UNIQUE)
1	Ronny	L1	NULL
2	Silvio	NULL	R1
3	Rocco	L2	R2
...

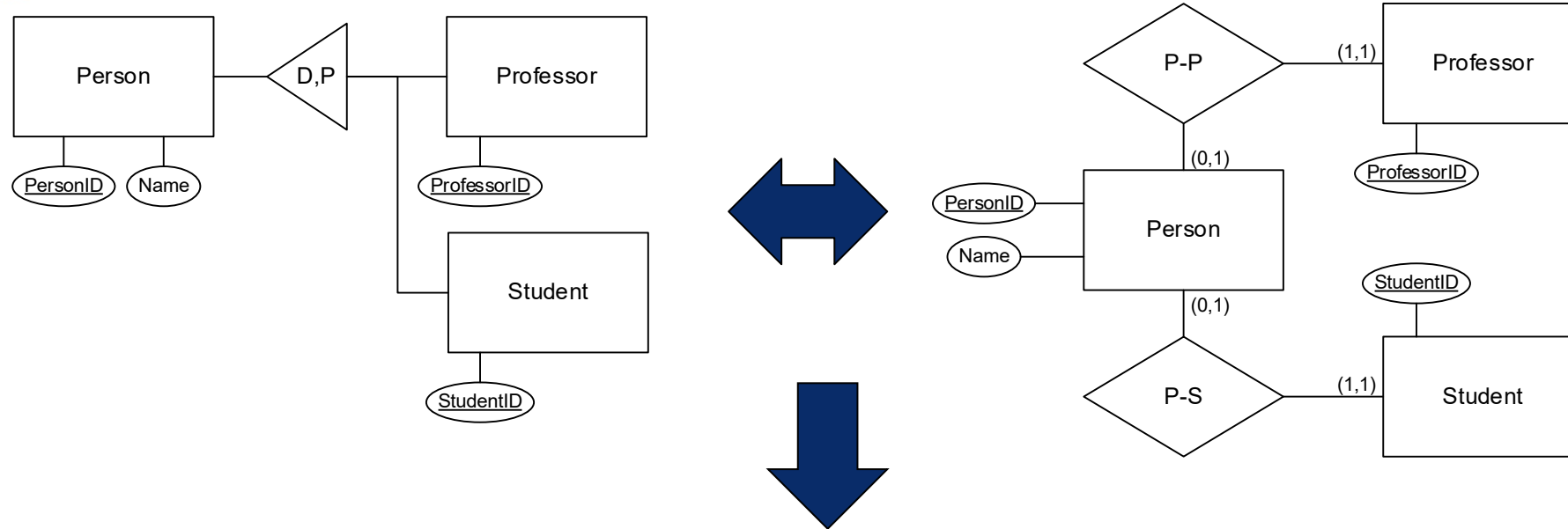
Relation <i>Linkshänder</i>	
<u>LinkshänderID</u>	...
L1	...
L2	...
...	...

Relation <i>Rechtshänder</i>	
<u>RechtshänderID</u>	...
R1	...
R2	...
...	...

..	NULL	NULL
----	------	------

Nicht disjunkt-Total: Jede Entität gehört *immer mindestens einem* Spezialfall an!

Generalisierung/ Spezialisierung (D,P)



Relation <i>Person</i>			
<u>PersonID</u>	Name	#ProfessorID (UNIQUE)	#StudentID (UNIQUE)
1	Ronny	NULL	S1
2	Silvio	NULL	S2
3	Christian	P1	NULL
...

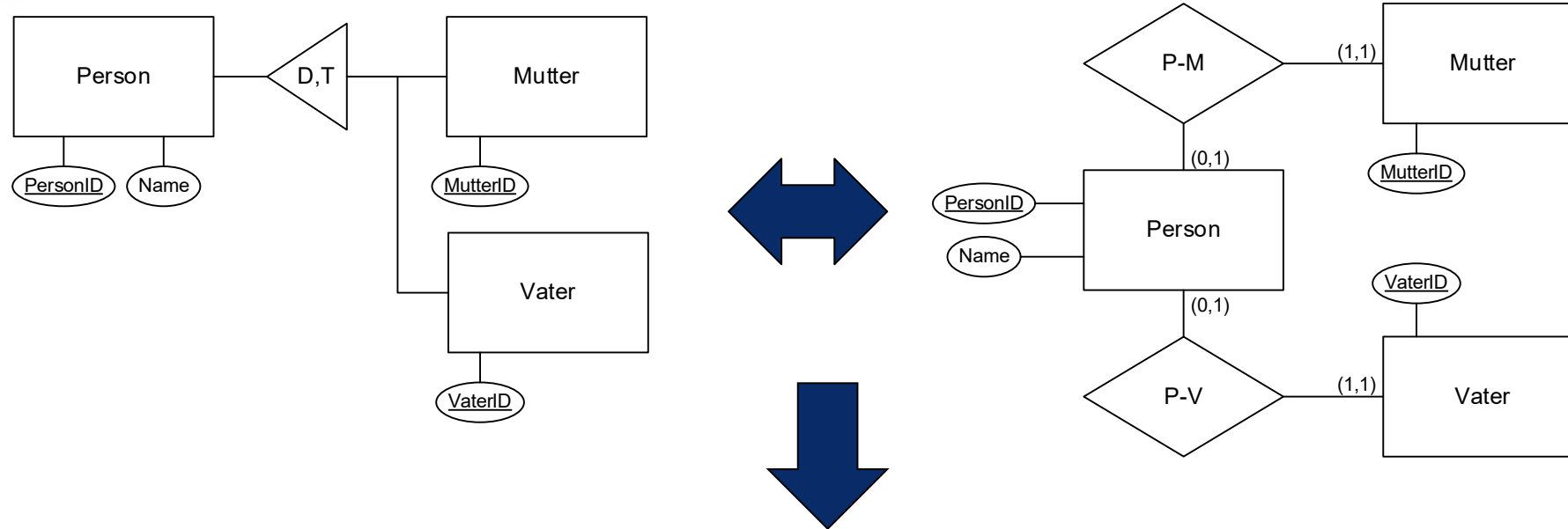
Relation <i>Professor</i>	
<u>ProfessorID</u>	...
P1	...
...	...

Relation <i>Student</i>	
<u>StudentID</u>	...
S1	...
S2	...
...	...

...	P2	S3
-----	----	----

Disjunkt-Partiell: Jede Entität kann *maximal einem* Spezialfall angehören!

Generalisierung/ Spezialisierung (D,T)



Relation <i>Person</i>			
<u>PersonID</u>	Name	#MutterID (UNIQUE)	#VaterID (UNIQUE)
1	Ronny	NULL	V1
2	Silvio	NULL	V2
3	Chantal	M1	NULL
...

Relation <i>Mutter</i>	
<u>MutterID</u>	...
M1	...
...	...

Relation <i>Vater</i>	
<u>VaterID</u>	...
V1	...
V2	...
V3	...
...	...

...	M2	V3
...	NULL	NULL

Disjunkt-Total: Jede Entität gehört *immer genau einem* Spezialfall an!