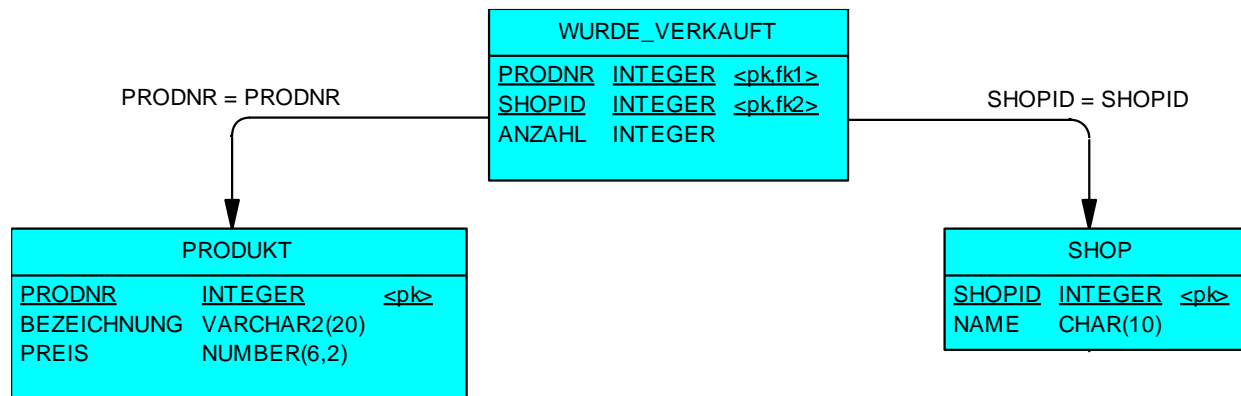




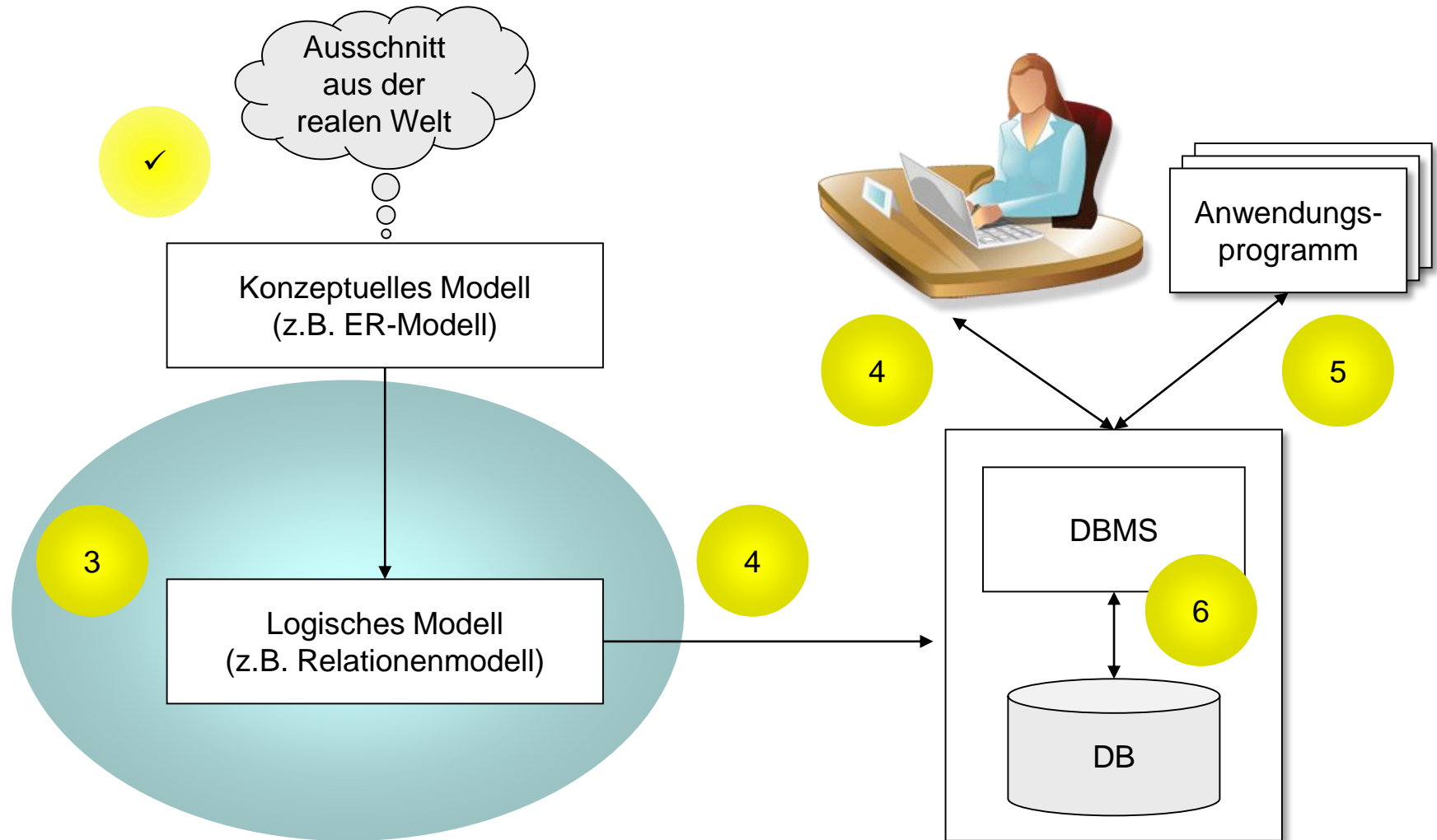
Datenbanken 1

– Kapitel 3: Relationenmodell –





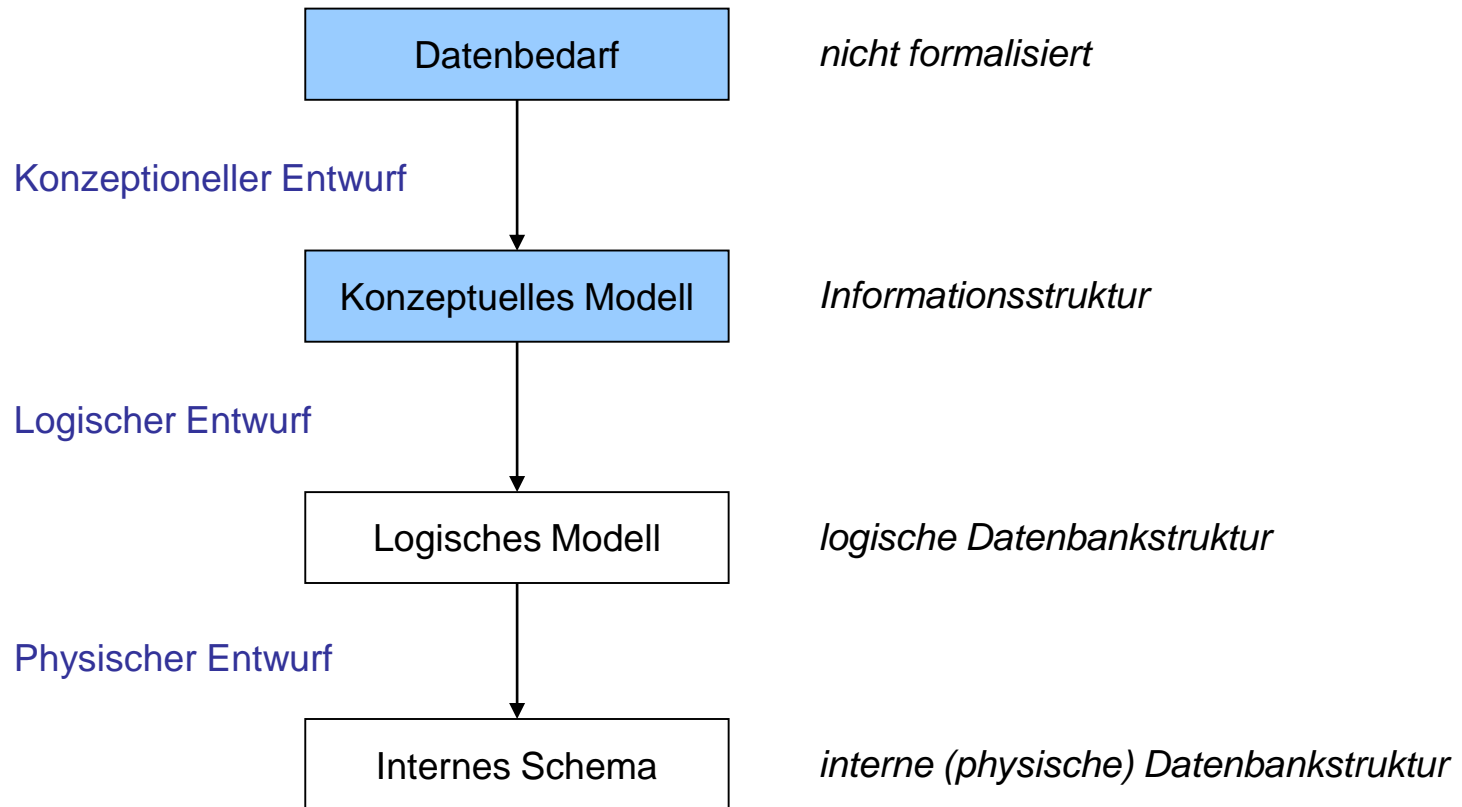
Vorlesung Datenbanken 1





Phasen des Datenbankentwurfs

Anforderungsanalyse





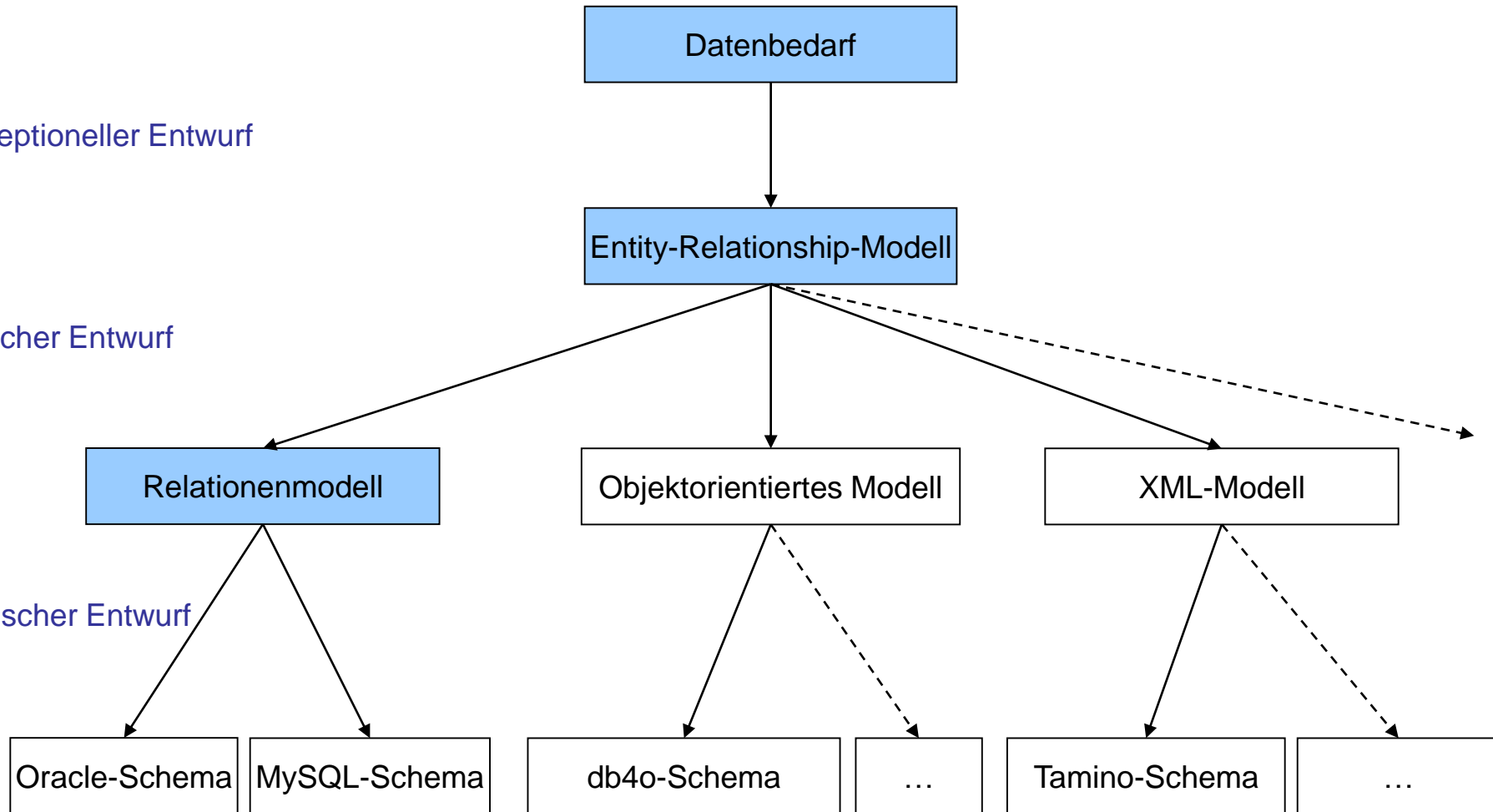
Datenbankentwurf – Beispiel

Anforderungsanalyse

Konzeptioneller Entwurf

Logischer Entwurf

Physischer Entwurf





Relationenmodell

Inhalt des Kapitels

- Grundlagen des Relationenmodell
- Abbildung des Entity-Relationship-Modells auf das Relationenmodell
- Normalformen

Lernziele

- Kennenlernen des Relationenmodells
- Selbständiges Entwerfen von relationalen Datenbankmodellen unter Berücksichtigung verschiedener Normalformen



Relationenmodell – Grundlagen

- entwickelt von E. F. Codd (1970)
- beruht auf dem mathematischen Begriff der **Relation**, den man anschaulich mit dem der Begriff **Tabelle** vergleichen kann
- alle Informationen sind in Relationen abgelegt

Definitionen

- Eine **n-stellige Relation R** ist definiert als Untermenge des kartesischen Produkts der Wertebereiche der zugehörigen Attribute A_1, A_2, \dots, A_n :

$$R(A_1, A_2, \dots, A_n) \subseteq W(A_1) \times W(A_2) \times \dots \times W(A_n)$$

Beispiel: Student (MatrNr, Name, Geburtsdatum) \subseteq integer \times string \times date

- Ein Element der Menge R wird als **Tupel** bezeichnet, d.h. $t \in R$

Beispiel: $t = (123456, \text{'Schmidt'}, 30.01.2014)$



Relationenmodell – Grundlagen

- Darstellungsmöglichkeit für R: n-spaltige Tabelle
- ⇒ Jede Relation kann als Tabelle dargestellt werden

PROF:	PNR	NAME	FB
	431326	Schütte	FBI
	174892	Erbs	FBI
	917384	Rebstock	FBW

- Relation ist eine Menge: Garantie der Eindeutigkeit der Tupel (Zeilen)
- ⇒ Primärschlüssel (und ggf. mehrere Schlüsselkandidaten)
- **Schlüssel:** minimale Menge von Attributen, deren Werte ein Tupel eindeutig identifizieren.



Relationales Datenbankmodell – Grundregeln*

1. Jede Zeile (Tupel) ist eindeutig und beschreibt ein Objekt der Miniwelt.
2. Die *Reihenfolge der Zeilen* ist *ohne Bedeutung*, d.h. durch ihre Reihenfolge wird keine für den Benutzer relevante Bedeutung ausgedrückt.
3. Die *Reihenfolge der Spalten* ist *ohne Bedeutung*, da sie einen eindeutigen Namen tragen.
4. Jeder Datenwert innerhalb einer Relation ist ein *atomares* Datenelement.
5. Alle für den Benutzer bedeutungsvollen Informationen sind ausschließlich durch Datenwerte ausgedrückt.
6. Es existieren ein Primärschlüssel und ggf. weitere Schlüsselkandidaten.

*E. F. Codd: A Relational Model of Data for Large Shared Data Banks. In: Commun. ACM 13(6): 377-387 (1970)



Bezeichnungen

Name des Relationenschemas

Attribut (Spalte)

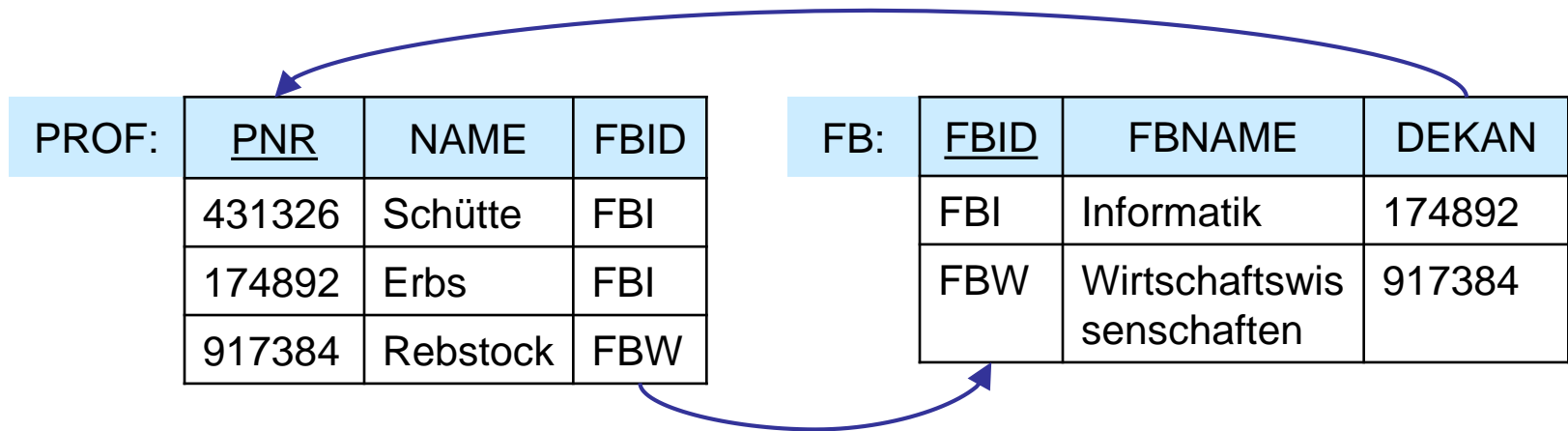
PROF:	<u>PNR</u>	NAME	FB	Relationenschema
Relation (Tabelleninhalt)	431326	Schütte	FBI	
	174892	Erbs	FBI	
	917384	Rebstock	FBW	Tupel (Zeile)

Datenbankschema: Menge von Relationenschemata



Integritätsbedingungen

- Integritätsbedingungen sind u.a. **Abhängigkeiten** zwischen Attributen - sowohl innerhalb einer Relation als auch zwischen Relationen.
- Eine Abhängigkeit innerhalb einer Relation nennt man **funktionale Abhängigkeit** zwischen Attributen.
Ein Spezialfall der funktionalen Abhängigkeit ist der **Schlüssel**.
- Abhängigkeiten zwischen Relationen:



⇒ Fremdschlüssel



Fremdschlüssel

- Ein **Fremdschlüssel (foreign key)** bezüglich einer Relation R_1 ist ein (ggf. zusammengesetztes) Attribut FK einer Relation R_2 , für das zu jedem Zeitpunkt gilt: zu jedem Wert (ungleich NULL) von FK muss ein gleicher Wert des Primärschlüssels PS (oder eines Schlüsselkandidaten SK) in irgendeinem Tupel von Relation R_1 sein.
- Eigenschaften
 - Fremdschlüssel und Primärschlüssel tragen wichtige interrelationale (oder intrarelationale) Informationen. Sie sind auf dem gleichen Wertebereich definiert.
 - Fremdschlüssel können Nullwerte aufweisen, wenn sie nicht selbst Teil des Primärschlüssels sind oder wenn nicht explizit NOT NULL deklariert ist.
 - Eine Relation kann mehrere Fremdschlüssel besitzen, welche die gleiche oder verschiedene Relationen referenzieren.
 - Referenzierte und referenzierende Relationen sind nicht notwendig verschieden (*self-referencing table*).



Transformation ER-Modell → Relationenmodell

- Entity-Relationship-Modell dient zur Modellierung der Realität aus Sicht der Anwendung
- Relationenmodell dient als Grundlage für die Realisierung in relationalen Datenbanken
- Transformation erfolgt durch eindeutige Regeln, mit deren Hilfe jedes Entity-Relationship-Modell in ein Relationenmodell überführt werden kann.
- Mit Hilfe von Case-Tools kann diese Transformation automatisiert werden.



Transformation ER-Modell → Relationenmodell

- Abbildung von Entity-Typen und einfachen Attributen
- Abbildung Beziehungstypen
 - o M:N Beziehungen
 - o 1:N Beziehungen
 - o 1:1 Beziehungen
 - o rekursive Beziehungen
 - o mehrstellige Beziehungen
- Abbildung schwacher Entity-Typen
- Abbildung mengenwertiger und komplexer Attribute
- Abbildung Generalisierung/Spezialisierung

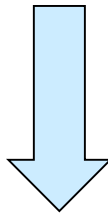
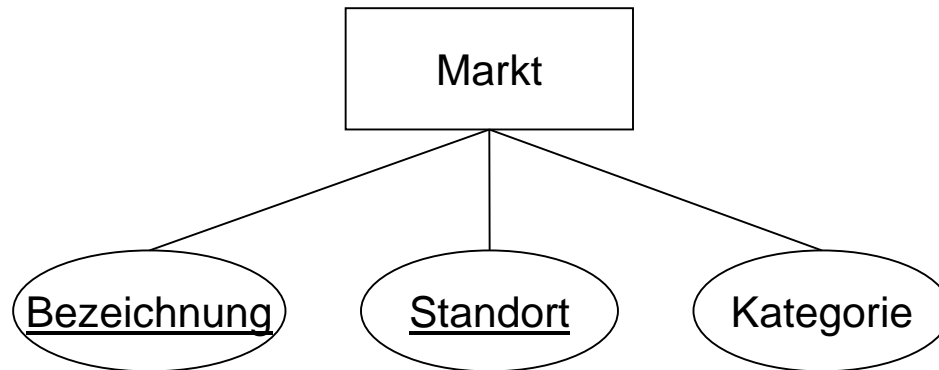


Abbildung von Entity-Typen

- jeder Entity-Typ wird zu einem Relationenschema
- einfache Attribute des Entity-Typs werden die Attribute des Relationenschemas
- falls Entities komplexe oder mengenwertig Attribute aufweisen, müssen diese aufgelöst werden (\Rightarrow Diskussion später)
- ein Schlüssel (falls noch nicht im ER-Modell geschehen) ist als Primärschlüssel des Relationenschemas auszuzeichnen, alternativ ist ein zusätzliches Schlüsselattribut hinzuzufügen
- die Datentypen zu den Attributen müssen definiert werden (falls noch nicht im ER-Modell geschehen)



Abbildung von Entity-Typen (Beispiel)

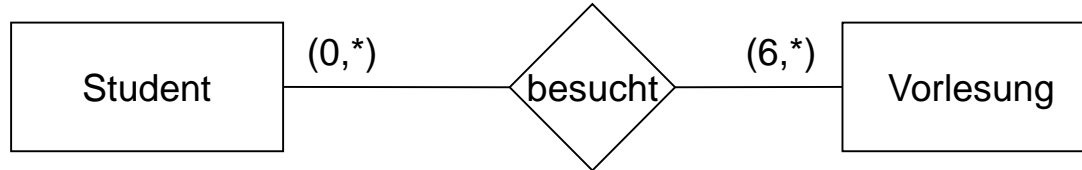


MARKT (BEZEICHNUNG, STANDORT, KATEGORIE)



Abbildung M:N Beziehungstypen

Abbildung M:N Beziehungstypen



Student (Matrikelnummer, Name)

Vorlesung (VorlesungsID, Bezeichnung)

- Jeder M:N Beziehungstyp wird zu einem eigenen Relationenschema
- Die Primärschlüssel der beteiligten Entity-Typen werden als Attribute hinzugefügt - diese bilden zusammen den Primärschlüssel und sind jeweils Fremdschlüssel bezogen auf die beiden aus den Entity-Typen entstandenen Relationenschemata
- Ggf. vorhanden Attribute des Beziehungstyps werden Attribute des Relationenschemas.

STUDENT (MATRIKELNUMMER, NAME)

VORLESUNG (VORLESUNGSID, BEZEICHNUNG)

BESUCHT (MATRIKELNUMMER → STUDENT, VORLESUNGSID → VORLESUNG)

Kardinalitäten?



Abbildung M:N Beziehungstypen

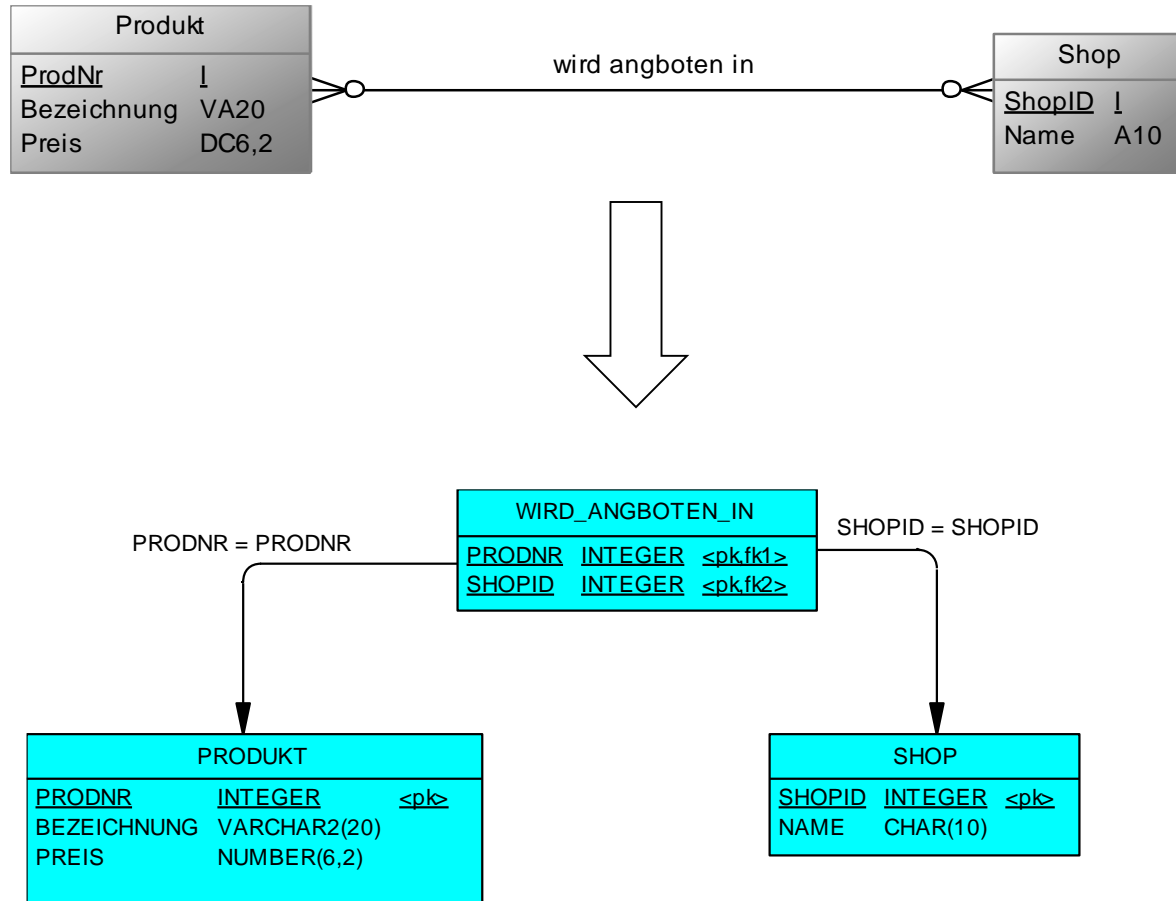




Abbildung M:N Beziehungstypen

- *alternative Darstellung:*

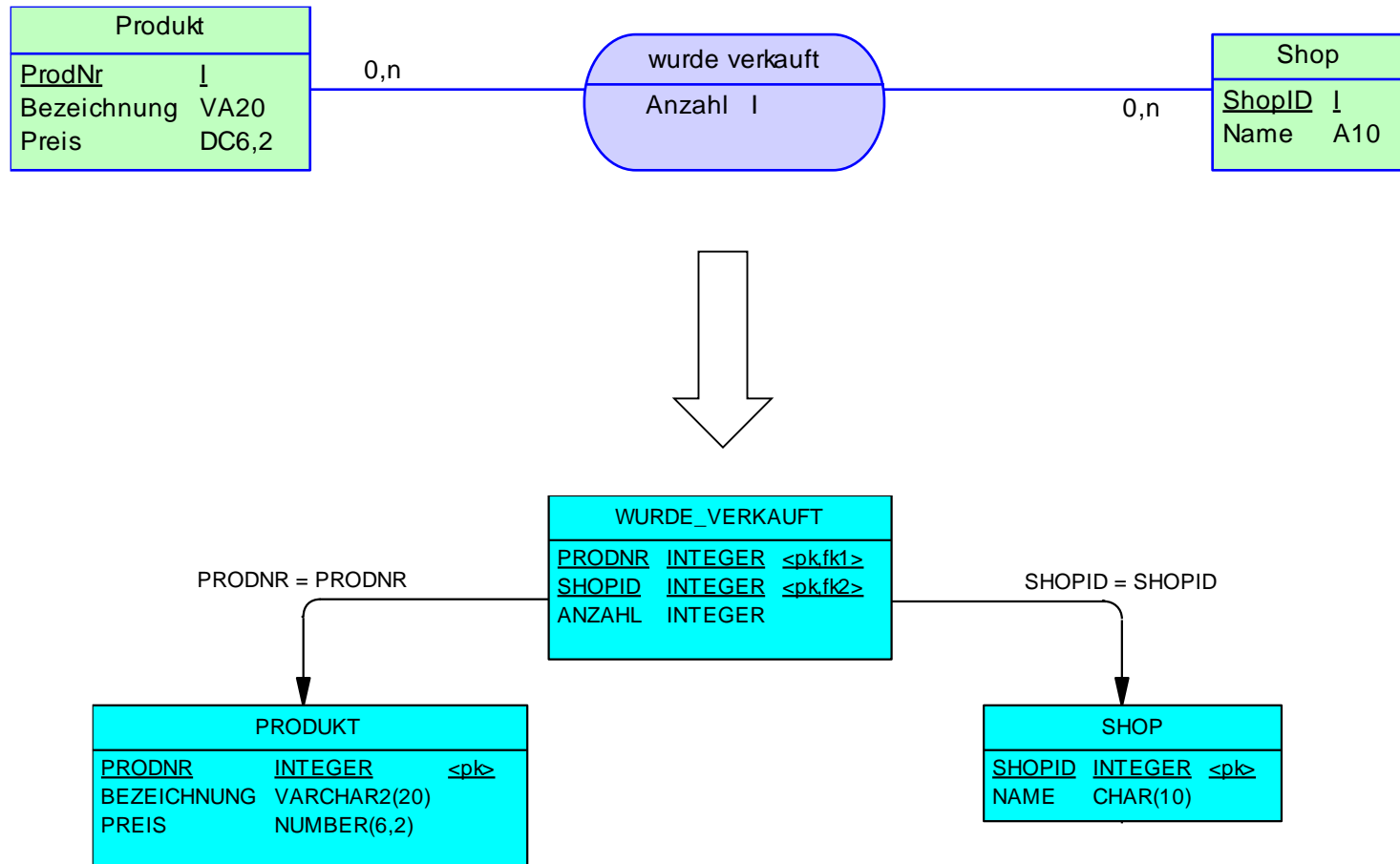




Abbildung M:N Beziehungstypen

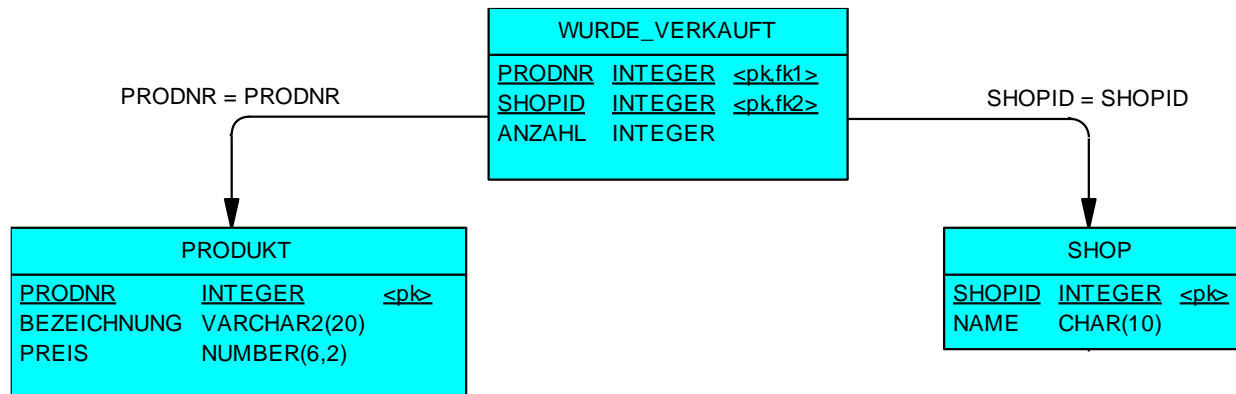
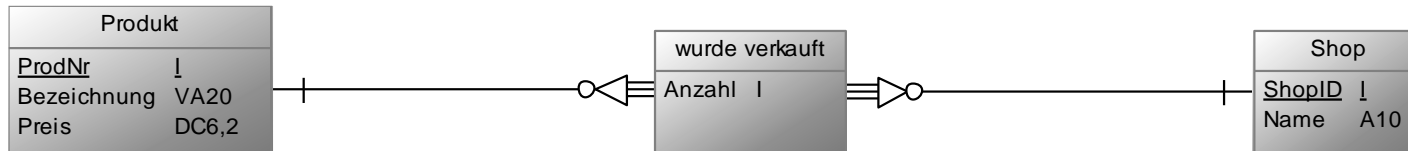
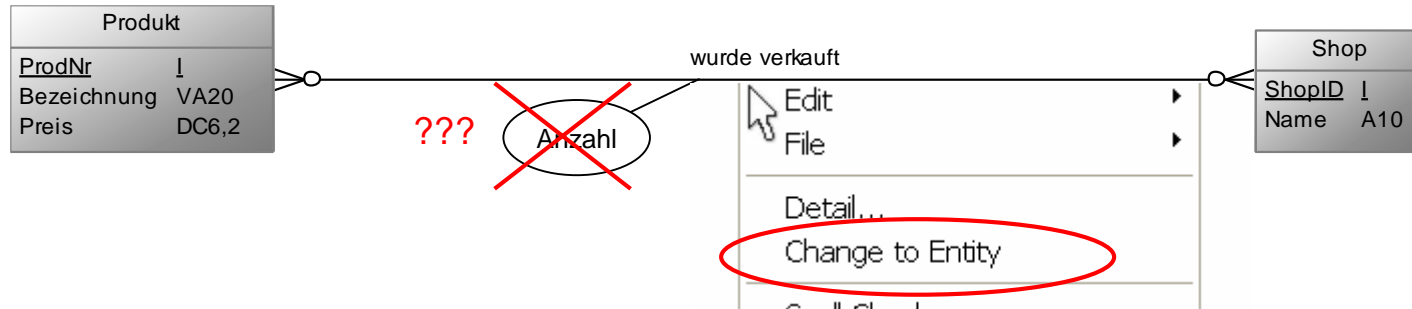


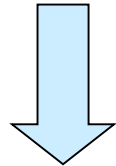


Abbildung 1:N Beziehungstypen

- Erster Ansatz: Vorgehen wie bei n:m Beziehungen



Mitarbeiter (PersNr, Name, Gehalt)
Abteilung (AbtName, Bereich)



MITARBEITER (PERSNR, NAME, GEHALT)

ABTEILUNG (ABTNAME, BEREICH)

GEHOERT_ZU (PERSNR → MITARBEITER, ~~ABTNAME~~ → ABTEILUNG)

GEHOERT_ZU (PERSNR → MITARBEITER, ABTNAME → ABTEILUNG)



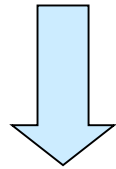
Abbildung 1:N Beziehungstypen

- Verfeinerung durch Zusammenfassung



Mitarbeiter (PersNr, Name, Gehalt)

Abteilung (AbtName, Bereich)



MITARBEITER (PERSNR, NAME, GEHALT, **ABTNAME → ABTEILUNG**)

ABTEILUNG (ABTNAME, BEREICH)

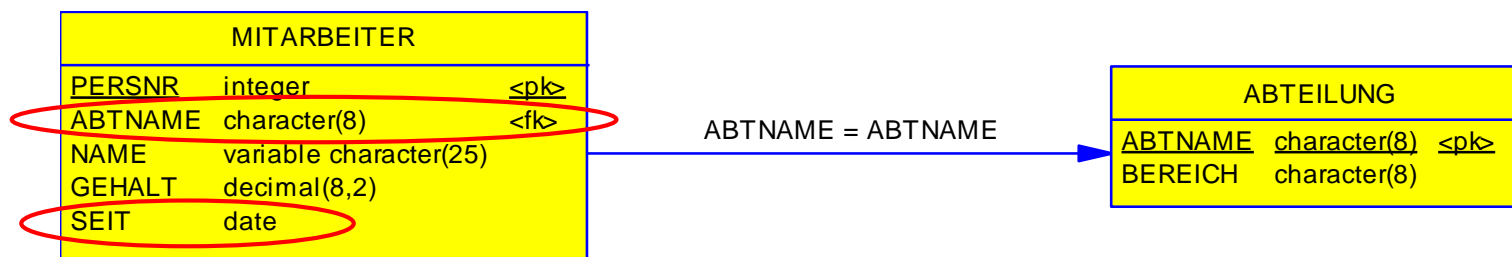
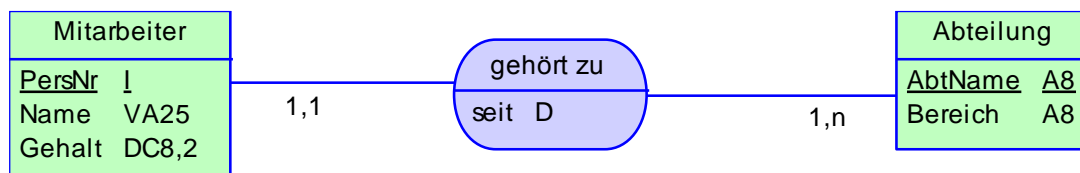
- Regel: Relationen mit gleichem Schlüssel kann man zusammenfassen
– aber nur diese und keine anderen!



Abbildung 1:N Beziehungstypen

Abbildung 1:N Beziehungen

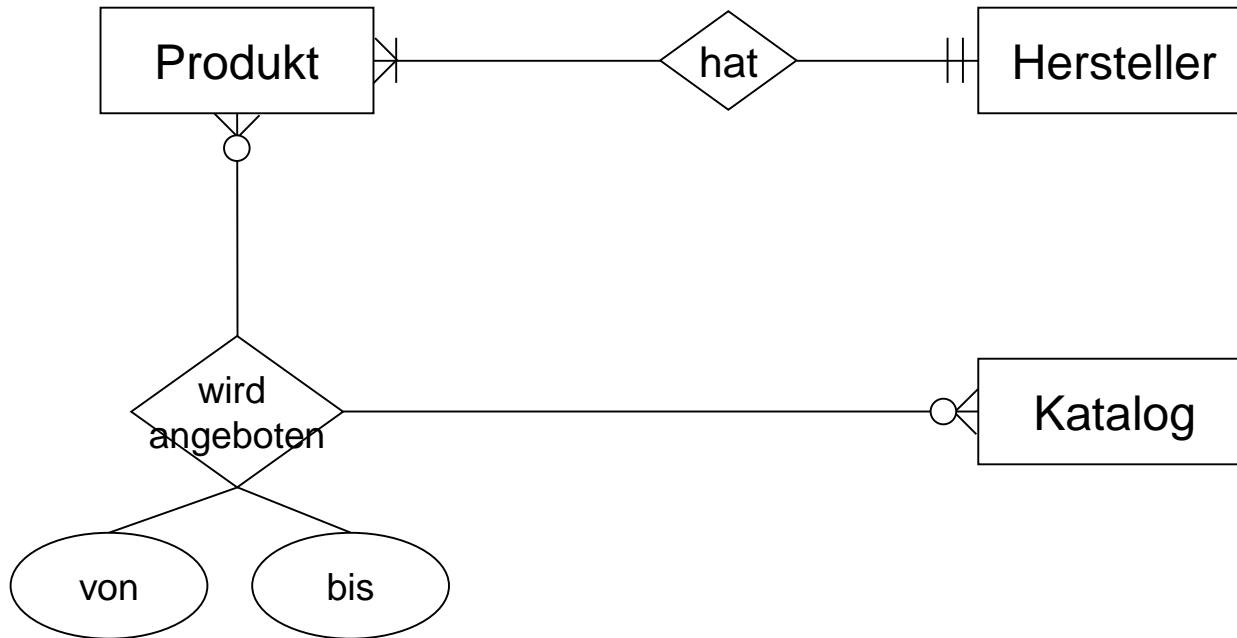
- Für 1:N Beziehungstypen wird **kein** zusätzliches Relationenschema angelegt!
- In das Relationenschema, dessen Tupel nur maximal ein Mal in der Beziehung erscheinen dürfen, wird der Primärschlüssel des anderen Relationenschemas als Fremdschlüssel hinzugefügt.
- Attribute der Beziehung werden ebenfalls in dem Relationenschema hinzugefügt, dessen Tupel nur ein Mal in der Beziehung erscheinen dürfen.





Hörsaalübung

Setzen Sie das folgende ER-Modell in ein Relationenschema um:



Produkt (Produktnummer, Bezeichnung, Preis)

Hersteller (HerstellerID, Herstellername)

Katalog (KatalogID, Katalogname)

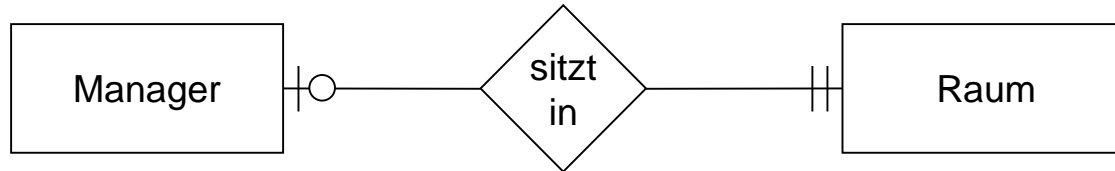


Transformation ER-Modell→Relationenmodell

- ✓ Abbildung von Entity-Typen und einfachen Attributen
- Abbildung Beziehungstypen
 - ✓ M:N Beziehungen
 - ✓ 1:N Beziehungen
 - o 1:1 Beziehungen
 - o rekursive Beziehungen
 - o mehrstellige Beziehungen
- Abbildung schwacher Entity-Typen
- Abbildung mengenwertiger und komplexer Attribute
- Abbildung Generalisierung/Spezialisierung



Abbildung 1:1 Beziehungstypen



Manager (PersNr, Name, Gehalt)

Raum (RaumNr, Quadratmeter)

- Mindestens einem der beiden Relationenschemata ist der Schlüssel des anderen als Fremdschlüssel hinzuzufügen (oder beiden):

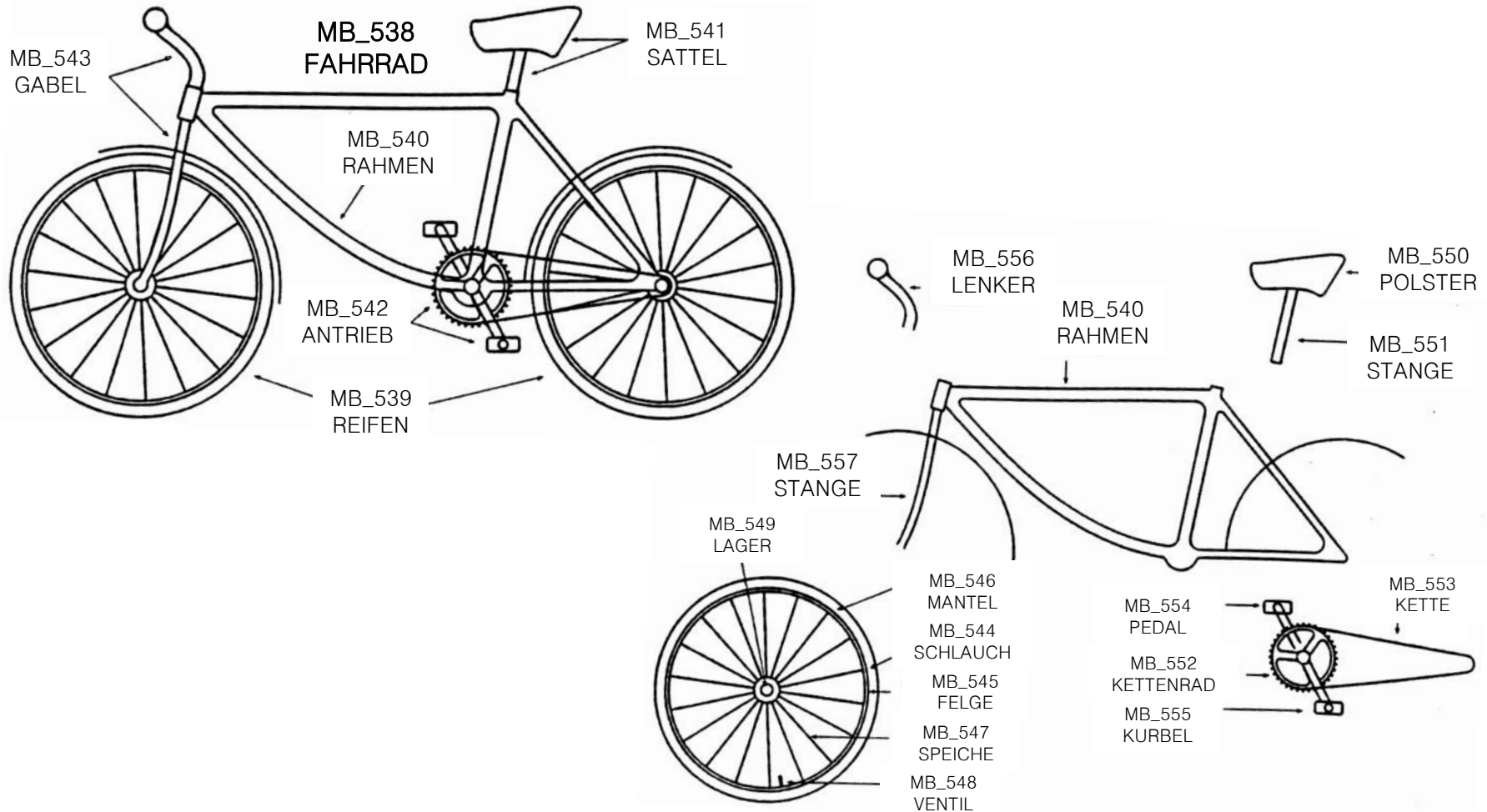
MANAGER (PERSNR, NAME, GEHALT, RAUMNR → RAUM)

RAUM (RAUMNR, QUADRATMETER, PERSNR → MANAGER)

- Anmerkung: Es könnten auch alle Attribute in ein Relationenschema aufgenommen werden, d. h. aus 2 Entities wird ein Relationenschema. Ggf. sinnvoll bei einer „echten“ 1:1 Beziehung (d.h. (1,1) und (1,1))

Abbildung rekursiver Beziehungstypen: Beispiel

Stückliste „Fahrrad“



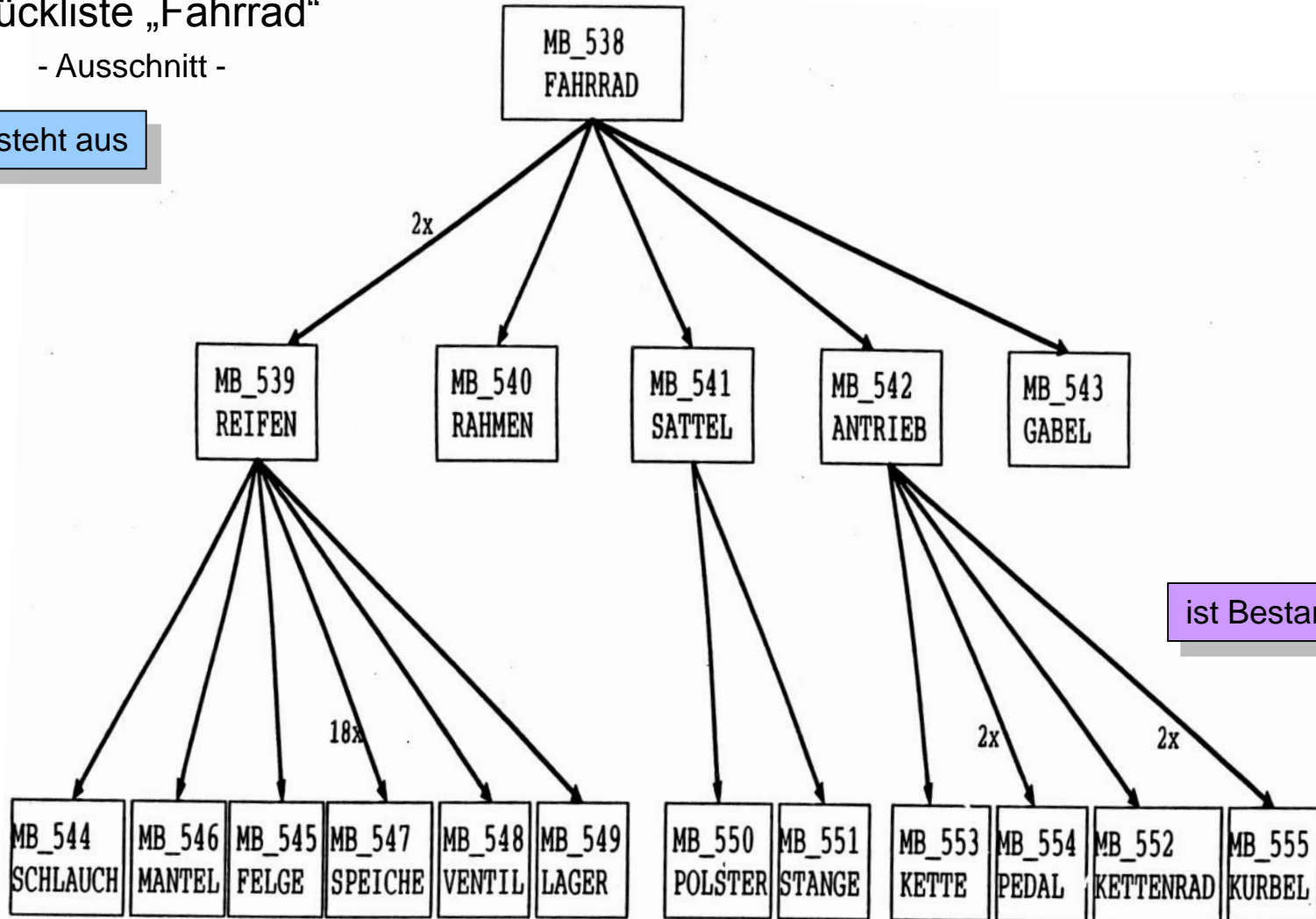
Quelle: I. Schestag, Datenbanken



Abbildung rekursiver Beziehungstypen: Beispiel

Stückliste „Fahrrad“
- Ausschnitt -

besteht aus



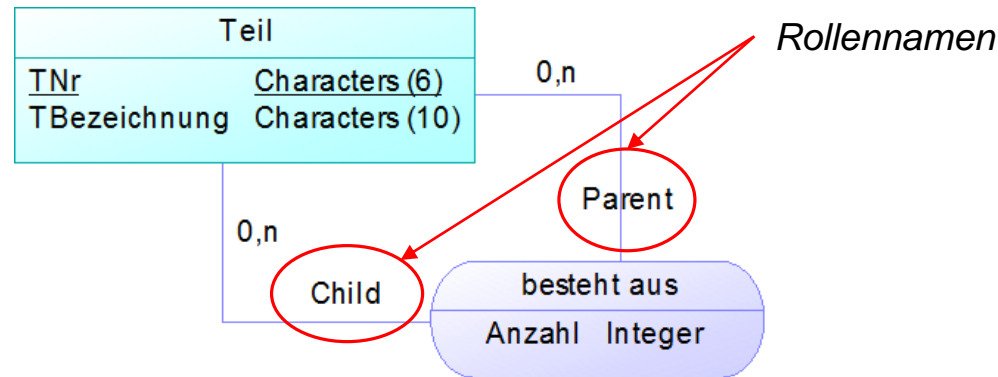
ist Bestandteil von

Quelle: I. Schestag, Datenbanken



Abbildung rekursiver Beziehungstypen

- Behandlung analog „nichtrekursiver“ Beziehungstypen



- ⇒ Aus dem Beziehungstyp entstehendes Relationenschemata enthält zwei Fremdschlüssel auf das aus dem Entity-Typ entstehende Relationenschemata – Namen anpassen!

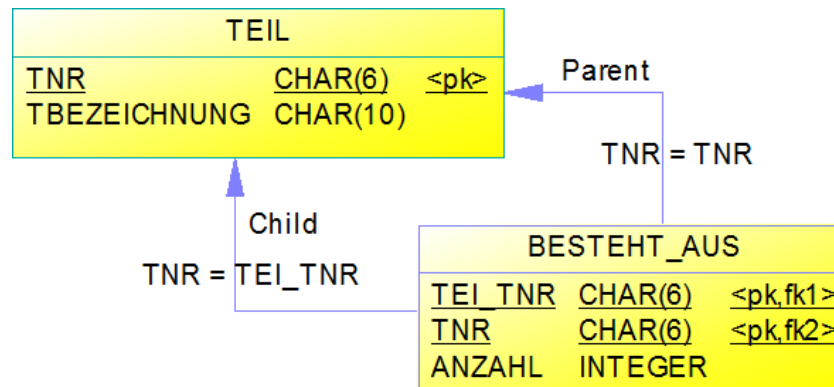




Abbildung rekursiver Beziehungstypen: Beispiel

TEIL

<u>TNR</u>	TBEZ	Einzelpreis
MB_538	Fahrrad	...
MB_539	Reifen	...
MB_540	Rahmen	...
MB_541	Sattel	...
MB_542	Antrieb	...
MB_543	Gabel	...
MB_544	Schlauch	...
MB_545	Felge	...
MB_546	Mantel	...
MB_547	Speiche	...
MB_548	Ventil	...
MB_549	Lager	...
...

BESTEHT_AUS

<u>TNR</u>	<u>TEI_TNR</u>	Anzahl
MB_538	MB_539	2
MB_538	MB_540	1
MB_538	MB_541	1
MB_538	MB_542	1
MB_539	MB_544	1
MB_539	MB_545	1
MB_539	MB_546	1
MB_539	MB_547	18
MB_539	MB_548	1
MB_539	MB_549	1
...

Quelle: I. Schestag, Datenbanken



Mehrstellige Beziehungen

- Für den Beziehungstyp wird ein eigenes Relationenschema angelegt, in welches die Primärschlüssel aller Beteiligten Entity-Typen als Fremdschlüssel übernommen werden; diese bilden zusammen den Primärschlüssel.
- Attribute des Beziehungstyps werden ebenfalls dem Relationenschema hinzugefügt.

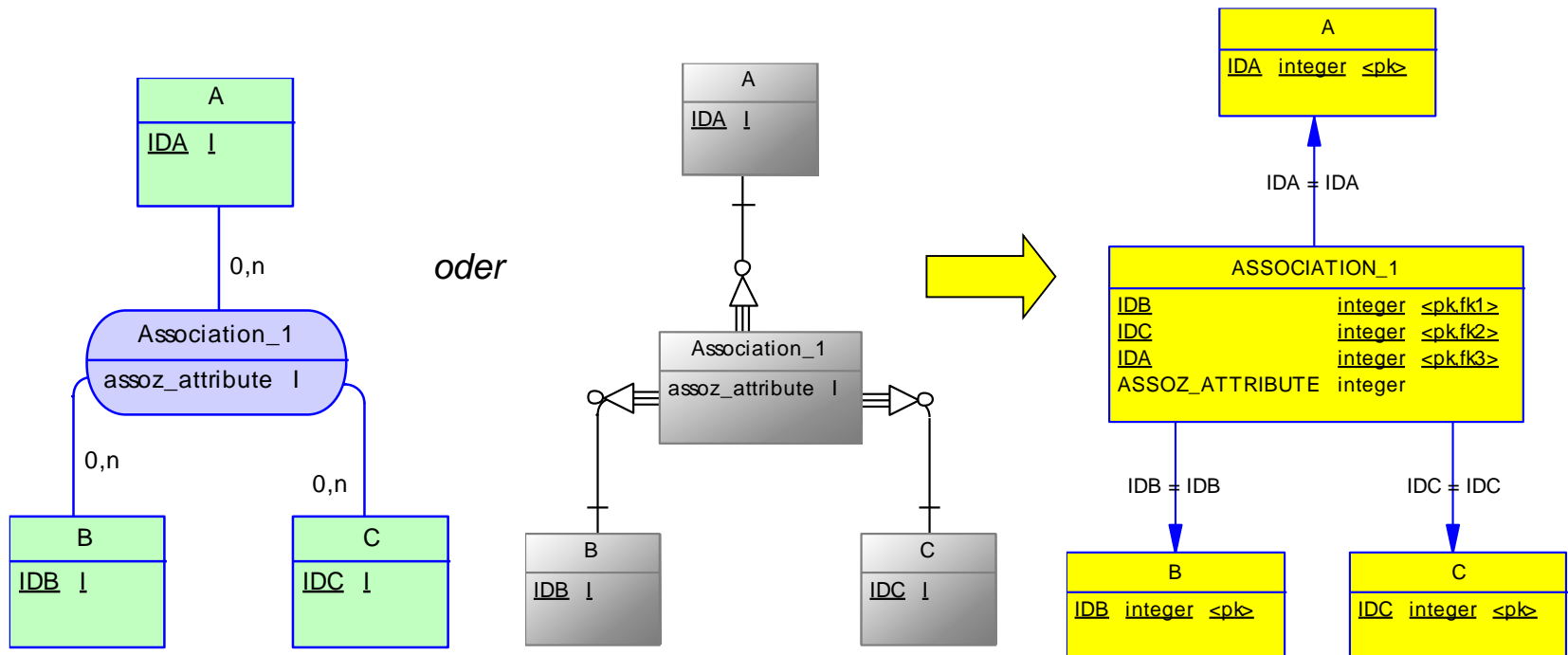
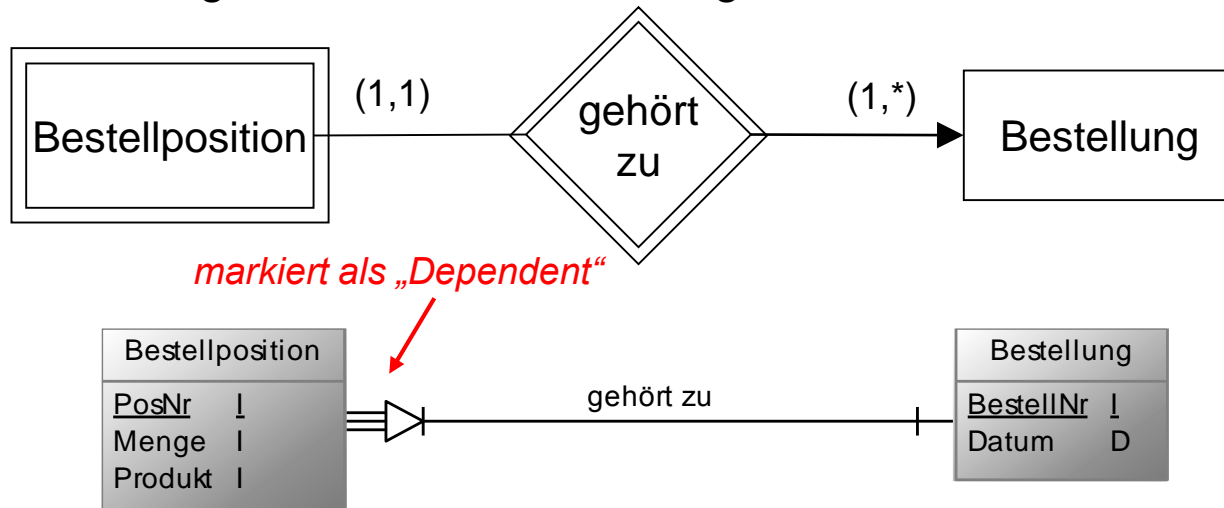




Abbildung schwacher Entity-Typen

- Zur Erinnerung: Funktionale Beziehung ist Bestandteil des Schlüssels



- Abweichend von „normalen“ 1:N Beziehungen, wird der Primärschlüssel nicht nur als Fremdschlüssel übernommen, sondern wird auch Bestandteil des Primärschlüssels auf der „N“-Seite.

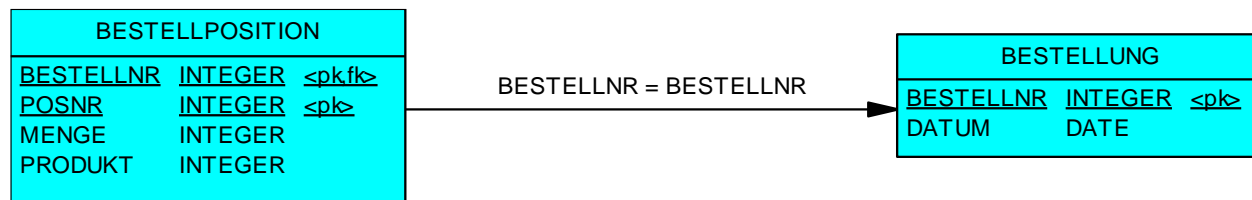
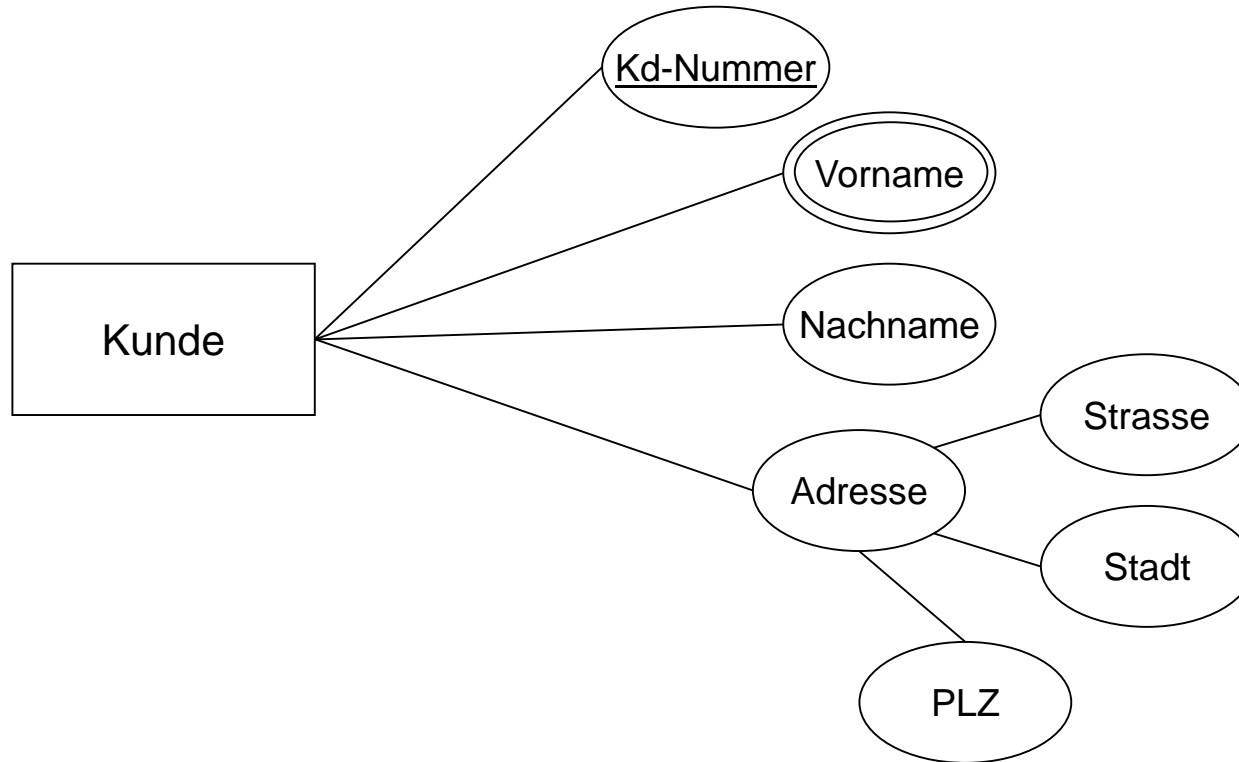




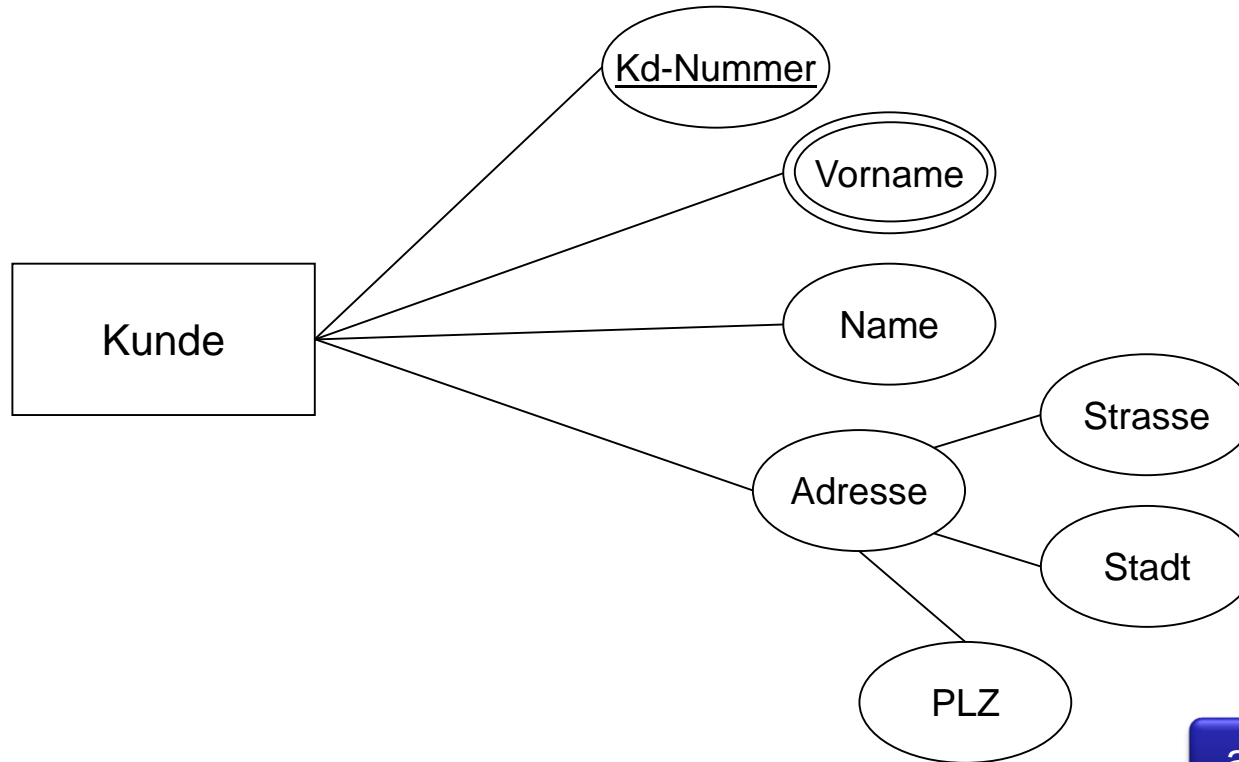
Abbildung mengenwertiger und strukturierter Attribute



~~KUNDE (KD-NUMMER, {VORNAME}, NACHNAME, ADRESSE (STRASSE, STADT, PLZ))~~



Abbildung mengenwertiger und strukturierter Attribute



andere Variante?

KUNDE (KD-NUMMER, NACHNAME, STRASSE, STADT, PLZ)

VORNAME (KD-NUMMER → KUNDE, VORNAME)



Abbildung mengenwertiger und strukturierter Attribute

- Anmerkung zu Case-Tools: Die Modellierung mengenwertiger und strukturierte Attribute wird von Case-Tools häufig nicht unterstützt.
- Lösungsvariante?



Kritik am Relationalen Modell

- Prinzip des Relationalen Modells führt dazu, dass oft zusammenhängende Inhalte auf mehrere Tabellen verteilt werden müssen.
 - ⇒ Performance-Verlust
 - ⇒ Entwicklung alternativer Ansätze
 - Objektorientierte Datenbanksysteme (kommend von OO-Sprachen, DER Datenbank-Hype der 90er Jahre)
 - Objektrelationale Datenbanksysteme (kommend von relationalen Datenbanksystemen – „Gegenreaktion“ der Hersteller relationaler DBMS Mitte/Ende der 90er Jahre)
 - Status Quo: die allermeisten Daten sind heute weltweit in Relationalen Datenbanksystemen gespeichert (und viele Daten auch noch in DBMS mit älteren Datenmodellen wie hierarchische Datenbanken – z.B. IMS) – für bestimmte Anwendungsszenarien Verwendung objektrelationaler Datenbanksysteme (z.B. Geodaten)
 - Aktueller Trend: NoSQL-Datenbanksysteme zur Speicherung hierarchischer Daten und mit flexiblem Schema für bestimmte Anwendungen

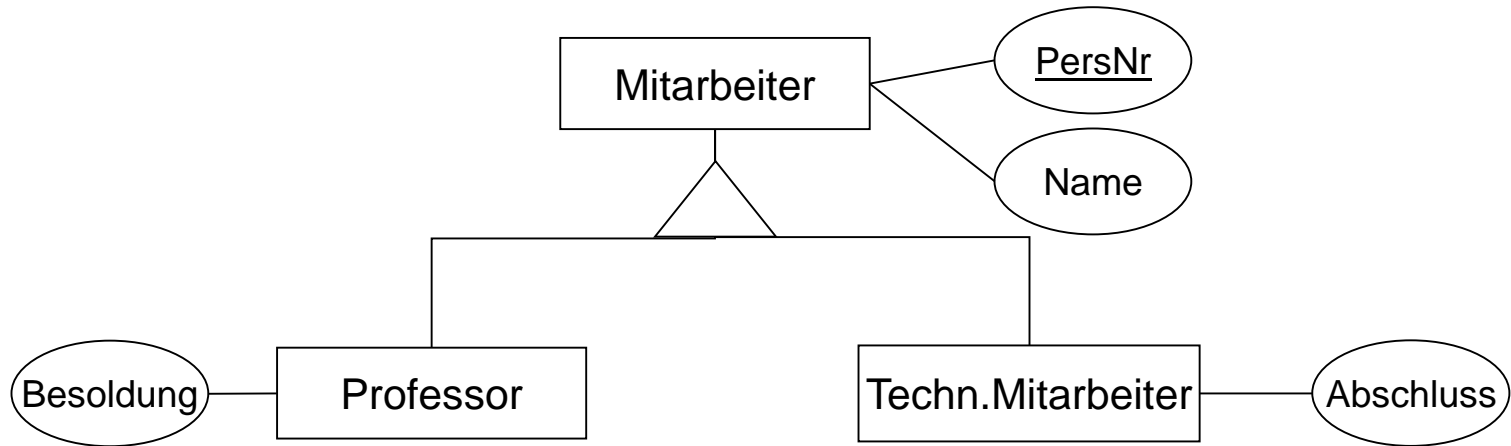


Transformation ER-Modell → Relationenmodell

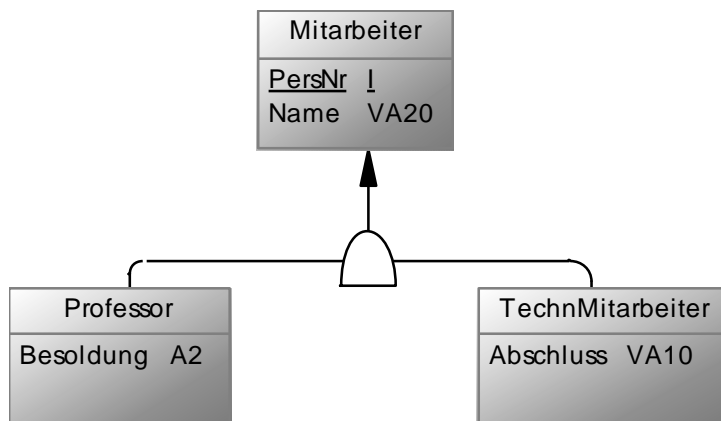
- ✓ Abbildung von Entity-Typen und einfachen Attributen
- ✓ Abbildung Beziehungstypen
 - ✓ M:N Beziehungen
 - ✓ 1:N Beziehungen
 - ✓ 1:1 Beziehungen
 - ✓ rekursive Beziehungen
 - ✓ mehrstellige Beziehungen
- ✓ Abbildung schwacher Entity-Typen
- ✓ Abbildung mengenwertiger und komplexer Attribute
- Abbildung Generalisierung/Spezialisierung



Abbildung Generalisierung/Spezialisierung



Darstellung im PowerDesigner:



Verschiedene Varianten der Abbildung ⇨



Abbildung Generalisierung/Spezialisierung

Variante 1: „Hausklassenmodell“

- Nur für die Spezialisierungen werden Relationenschemata ausgeprägt.

PROFESSOR		
<u>PERSNR</u>	integer	<pk>
BESOLDUNG	character(2)	
NAME	variable character(20)	

TECHNMITARBEITER		
<u>PERSNR</u>	integer	<pk>
ABSCHLUSS	variable character(10)	
NAME	variable character(20)	

PROF:	<u>PERSNR</u>	NAME	BESOLDUNG
	0814	Beckstein	C3
	0815	Küspert	C4

TM:	<u>PERSNR</u>	NAME	ABSCHLUSS
	0665	Friedel	Dr. rer. nat.
	0666	Mäurer	Dipl.-Math.

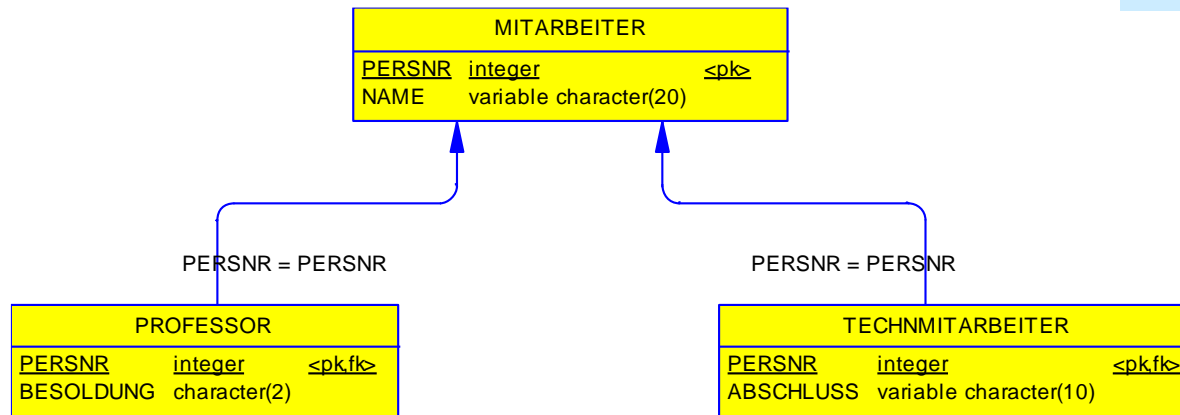
- Vorteile?
- Nachteile?



Abbildung Generalisierung/Spezialisierung

Variante 2: „Partitionierungsmodell“

- Sowohl für die Spezialisierungen als auch die Generalisierung werden Relationenschemata ausgeprägt.
- In den Relationenschemata der Spezialisierungen werden die Primärschlüssel der Generalisierung als Fremdschlüssel (und gleichzeitig Primärschlüssel) übernommen.



MITARBEITER:	<u>PERSNR</u>	NAME
	0665	Friedel
	0666	Mäurer
	0814	Beckstein
	0815	Küspert

PROF:	<u>PERSNR</u>	BESOLDUNG
	0814	C3
	0815	C4

TM:	<u>PERSNR</u>	ABSCHLUSS
	0665	Dr. rer. nat.
	0666	Dipl.-Math.

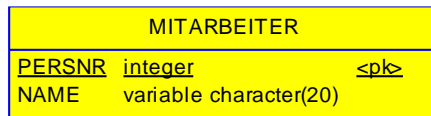
- Vorteile?
- Nachteile?



Abbildung Generalisierung/Spezialisierung

Variante 3: „Volle Redundanz“

- Sowohl für die Spezialisierungen als auch die Generalisierung werden Relationenschemata ausgeprägt
- In den Relationenschemata der Spezialisierungen werden alle Attribute der Generalisierung übernommen und die übernommenen Primärschlüssel als Fremdschlüssel (und Primärschlüssel) definiert.

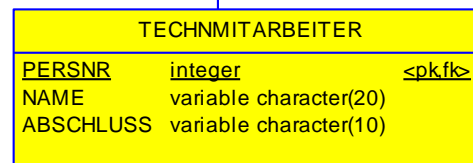
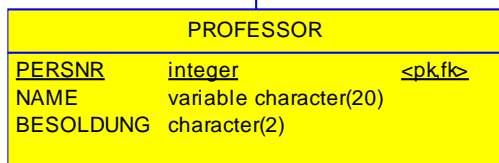


MITARBEITER:

<u>PERSNR</u>	NAME
0665	Friedel
0666	Mäurer
0814	Beckstein
0815	Küspert

PERSNR = PERSNR

PERSNR = PERSNR



PROF:

<u>PERSNR</u>	NAME	BESOLDUNG
0814	Beckstein	C3
0815	Küspert	C4

- Vorteile?
- Nachteile?

TM:

<u>PERSNR</u>	NAME	ABSCHLUSS
0665	Friedel	Dr. rer. nat.
0666	Mäurer	Dipl.-Math.



Abbildung Generalisierung/Spezialisierung

Variante 4: „Hierarchierelation“

- Es wird nur ein Relationenschema für die Generalisierung ausgeprägt
- Zusätzliches Attribut zur Typidentifikation
- Nullwerte für Attribute, welche in der zugehörigen Klasse nicht vorhanden sind.

MITARBEITER		
<u>PERSNR</u>	<u>integer</u>	<u><pk></u>
NAME	variable character(20)	
ABSCHLUSS	variable character(10)	
BESOLDUNG	character(2)	

MITARBEITER:	<u>PERSNR</u>	TYP	NAME	ABSCHLUSS	BESOLDUNG
	0665	TechnMit	Friedel	Dr. rer. nat.	-
	0666	TechnMit	Mäurer	Dipl.-Math.	-
	0814	Professor	Beckstein	-	C3
	0815	Professor	Küspert	-	C4

- Vorteile?
- Nachteile?

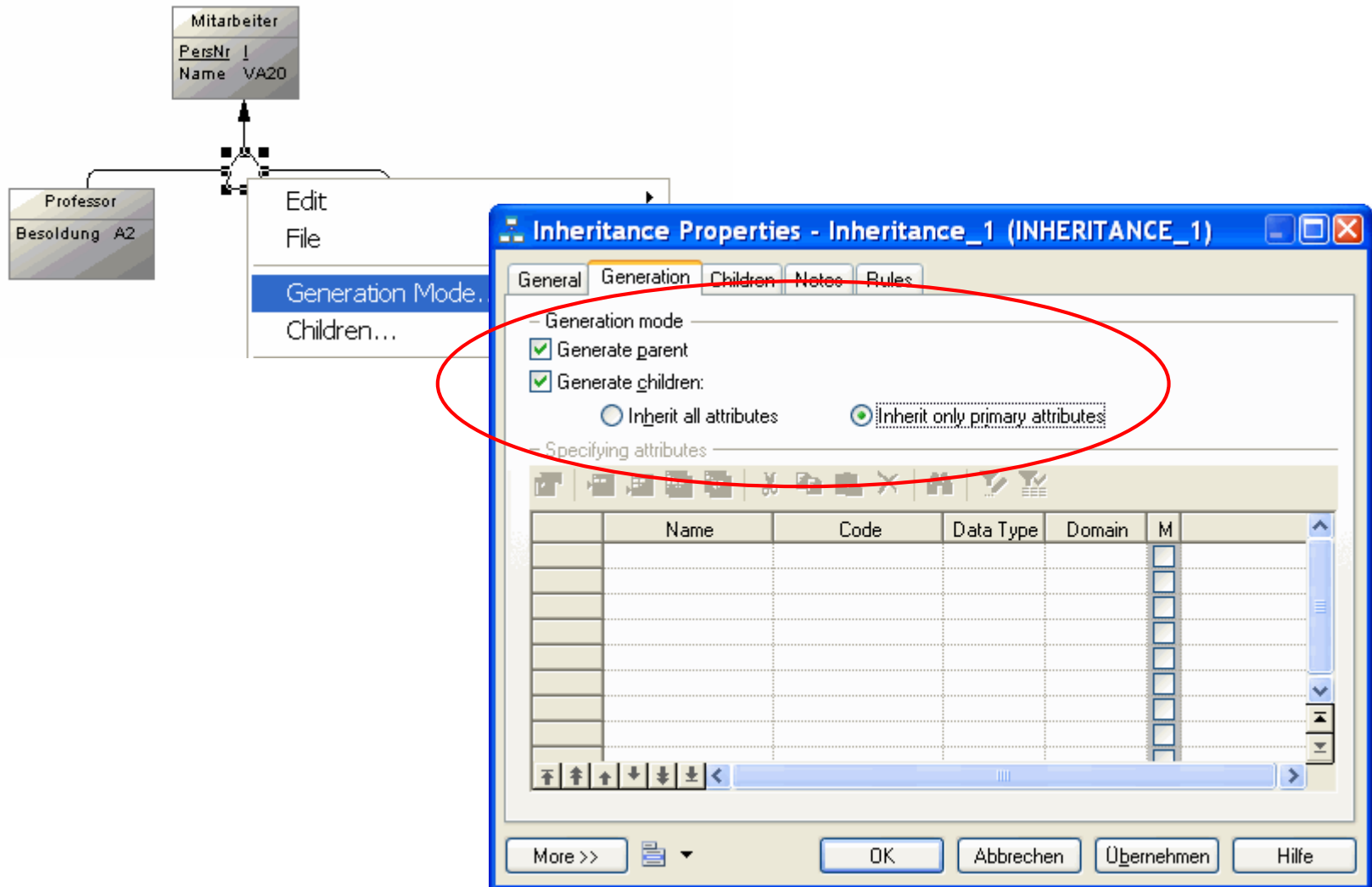


Abbildung Generalisierung / Spezialisierung

- Vier verschiedene Varianten der Abbildung
- Vor- und Nachteile bezüglich
 - Performance beim Zugriff auf generalisierte / spezialisierte Daten
 - Beziehungen zu anderen Klassen
 - Datenredundanz
 - Speicherbedarf



Abbildung Generalisierung/Spezialisierung





Zusammenfassung

- Relation (Tabelle)
 - alle Informationen werden ausschließlich durch atomare Werte dargestellt
 - Integritätsbedingungen werden auf/zwischen Relationen definiert
 - Referentielle Integrität als *wertebasierte* Beziehung
 - Kardinalitätsrestriktionen außer 0, 1 und * können nicht abgebildet werden
- Abbildung von Beziehungen durch Primärschlüssel/Fremdschlüssel
 - Alle Beziehungstypen müssen durch 1:N Beziehungen dargestellt werden
 - ⇒ M:N Beziehungstypen werden durch eigene Relationenschemata abgebildet
 - ⇒ 1:N Beziehungstypen müssen nicht als eigene Relationenschema abgebildet werden
 - ⇒ 1:1 Beziehungstypen müssen nicht als eigene Relationenschema abgebildet werden; u.U. ist eine Zusammenfassung der beiden aus den Entity-Typen entstandenen Relationenschemata sinnvoll.
 - Verschiedene Abbildungsmöglichkeiten für Generalisierung/Spezialisierung (Konzept nicht wirklich „kompatibel“ mit Relationenmodell)



Kapitel 3: Relationenmodell

- ✓ Grundlagen des Relationenmodell
- ✓ Abbildung des Entity-Relationship-Modells auf das Relationenmodell
- Normalformen



Normalisierung von Relationenschemata

Ziel/Motivation

- Vermeidung von Anomalien in Relationenschemata
- Anomalien:
 - Zustände in relationalen Datenbanken, in denen die Veränderung von Daten zu Datenbankzuständen führt, die nicht die gewünschte Realität darstellt
 - Unterscheidung zwischen Einfüge-, Änderungs- und Lösch-Anomalie

Weg

- Vermeidung von Anomalien in Relationenschemata wird erreicht durch systematische Vorgehensweise beim konzeptionellen Datenbankentwurf (ERM) und bei der Abbildung zum Relationalen Modell
- Entfernung von Anomalien ist nötig, wenn nicht systematisch modelliert wurde



Normalformen: Übersicht

Es existieren verschiedene Normalformen, welche jeweils aufeinander aufbauen (d.h. jede Normalform fordert, dass die vorhergehende Normalform erfüllt ist):

- 1NF (1. Normalform)
 - 2NF (2. Normalform)
 - 3NF (3. Normalform)
-
- BCNF (Boyce-Codd-Normalform)
 - 4NF (4. Normalform)
 - 5NF (5. Normalform)

Praktisch relevant sind insbesondere die ersten drei Normalformen!



Erste Normalform (1NF)

Eine Relationenschema ist in **erster Normalform**, wenn alle Attribute des Schemas elementar sind.

- ⇒ Für Attributwerte sind nur einfache Datentypen erlaubt, z.B. integer, real, string etc.
- ⇒ Listen, Mengen oder Relationen als Attribute sind nicht erlaubt (z.B. record- oder array-Typen).
- ➔ Entspricht der bisher verwendeten Definition des relationalen Modells



Einfüge-Anomalie (Insert-Anomalie)

Toepferprodukt_Markt

<u>Prod-Nr</u>	Produktart	Funktion	<u>Verkaufsmarkt</u>	Marktstandort	marktspez.Preis
11022	Tee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
10622	Kaffee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
20131	Schale	Deko	Rheinischer Tonmarkt	Mainz	80 €
20131	Schale	Deko	Odenwälder Töpferwelt	Erbach	50 €
20131	Schale	Deko	Internat. Tonmarkt	Strasbourg	120 €
40030	Krug	Deko	Internat. Tonmarkt	Strasbourg	100 €
40031	Krug	Deko	Odenwälder Töpferwelt	Erbach	80 €

- Ein neues Produkt wird eingeführt, aber noch nicht auf den Markt gebracht
- ⇒ Einfügen Produkt (33033, Schüssel, Gebrauch)
- Problem?
 - Ursache?



Änderungs-Anomalie (Update-Anomalie)

Toepferprodukt_Markt

<u>Prod-Nr</u>	Produktart	Funktion	<u>Verkaufsmarkt</u>	Marktstandort	marktspez.Preis
11022	Tee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
10622	Kaffee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
20131	Schale	Deko	Rheinischer Tonmarkt	Mainz	80 €
20131	Schale	Deko	Odenwälder Töpferwelt	Erbach	50 €
20131	Schale	Deko	Internat. Tonmarkt	Strasbourg	120 €
40030	Krug	Deko	Internat. Tonmarkt	Strasbourg	100 €
40031	Krug	Deko	Odenwälder Töpferwelt	Erbach	80 €

- Der Verkaufsmarkt „Odenwälder Töpferwelt“ wird von Erbach nach Michelstadt verlegt
- Problem?
- Ursache?



Lösch-Anomalie (Delete-Anomalie)

Toepferprodukt_Markt

<u>Prod-Nr</u>	Produktart	Funktion	<u>Verkaufsmarkt</u>	Marktstandort	marktspez.Preis
11022	Tee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
10622	Kaffee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
20131	Schale	Deko	Rheinischer Tonmarkt	Mainz	80 €
20131	Schale	Deko	Odenwälder Töpferwelt	Erbach	50 €
20131	Schale	Deko	Internat. Tonmarkt	Strasbourg	120 €
40030	Krug	Deko	Internat. Tonmarkt	Strasbourg	100 €
40031	Krug	Deko	Odenwälder Töpferwelt	Erbach	80 €

- Produkt 20131 soll aus dem Programm genommen werden
⇒ Löschen des Tupels mit der Prod-Nr 20131
- Problem?
- Ursache?



Ursachen von Anomalien

- **Redundante Datenhaltung:**

- Beispiele:

- jedes Produkt ist mehrfach mit allen Attributen abgespeichert
 - jeder Verkaufsmarkt ist mehrfach mit allen Attributen abgespeichert.

- **Ungünstige funktionale Abhängigkeiten:**

- Beispiel:

- Produktart hängt funktional nur von der Produkt-Nr ab, aber nicht von Verkaufsmarkt (welcher ebenfalls Bestandteil des Schlüssels ist)



Funktionale Abhängigkeit

- Funktionale Abhängigkeit:
 - Eine Menge Y von Attributen $\{y_1, y_2, \dots, y_n\}$ ist *funktional abhängig* von einer Menge X von Attributen $\{x_1, x_2, \dots, x_n\}$, wenn es eine Funktion zwischen X und Y gibt, d.h. für alle $\{x_1, x_2, \dots, x_n\}$ Elemente aus X gibt es genau ein $\{y_1, y_2, \dots, y_n\}$ aus Y .
- Mit anderen Worten:
 - In einer Relation ist eine Attribut(-kombination) Y funktional abhängig von einer Attribut(-kombination) X , wenn für gleiche X -Werte jeweils gleiche Y -Werte vorhanden sind, d.h. unterscheiden sich zwei Tupel in den X -Attributen nicht, so haben sie auch gleiche Werte für alle Y -Attribute
- Notation für funktionale Abhängigkeit (*functional dependency, FD*)
 - $X \rightarrow Y$ bzw. $\{x_1, x_2, \dots, x_n\} \rightarrow \{y_1, y_2, \dots, y_n\}$



Funktionale Abhängigkeit – Beispiel

Toepferprodukt_Markt

<u>Prod-Nr</u>	Produktart	Funktion	<u>Verkaufsmarkt</u>	Marktstandort	marktspez.Preis
11022	Tee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
10622	Kaffee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
20131	Schale	Deko	Rheinischer Tonmarkt	Mainz	80 €
20131	Schale	Deko	Odenwälder Töpferwelt	Erbach	50 €
20131	Schale	Deko	Internat. Tonmarkt	Strasbourg	120 €
40030	Krug	Deko	Internat. Tonmarkt	Strasbourg	100 €
40031	Krug	Deko	Odenwälder Töpferwelt	Erbach	80 €

T_M (PRODNR, PRODART, FUNKTION, VERKAUFSMARKT, MARKSTANDORT, MARKTSPEZPREIS)

- Funktionale Abhängigkeiten:



Funktionale Abhängigkeit - Schlüssel

Formalisierung des Schlüsselbegriffs:

- ⇒ Konzept der vollen funktionalen Abhängigkeit:
Eine Menge Y von Attributen $\{y_1, y_2, \dots, y_n\}$ ist **voll funktional abhängig** von einer Menge X von Attributen $\{x_1, x_2, \dots, x_n\}$, wenn
 - Y funktional abhängig von X ist, d.h. $X \rightarrow Y$ und
 - X nicht mehr verkleinert werden kann, d.h. für alle $\forall x_i \in X: X - \{x_i\} \not\rightarrow Y$
- ⇒ Falls Relation R voll funktional abhängig von X , so bezeichnet man X als *Schlüsselkandidat* von R .
- ⇒ Im allgemeinen wird aus den Schlüsselkandidaten der Primärschlüssel ausgewählt.



Zweite Normalform (2NF)

Eine Relationenschema ist in **zweiter Normalform**, wenn

- es in erster Normalform ist und
- jedes Nichtschlüsselattribut *voll funktional* von jedem Schlüsselkandidat abhängt (und nicht nur von einem Teilschlüssel).

Bemerkungen

- Abhängigkeiten von einem Teil des Schlüssels (bei zusammengesetzten Schlüsseln) führen zur Anomalien.
- Intuitive Formulierung: ein Relationenschema verletzt die zweite Normalform (2NF), wenn in der Relation Informationen über mehr als ein Konzept modelliert werden.
- Hinweis: Relationenschemata, die in 1NF sind und deren Schlüssel aus einem Attribut bestehen, sind in 2NF (folgt aus der Definition).



Verletzung der 2NF

Toepferprodukt_Markt

<u>Prod-Nr</u>	Produktart	Funktion	<u>Verkaufsmarkt</u>	Marktstandort	marktspez.Preis
11022	Tee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
10622	Kaffee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
20131	Schale	Deko	Rheinischer Tonmarkt	Mainz	80 €
20131	Schale	Deko	Odenwälder Töpferwelt	Erbach	50 €
20131	Schale	Deko	Internat. Tonmarkt	Strasbourg	120 €
40030	Krug	Deko	Internat. Tonmarkt	Strasbourg	100 €
40031	Krug	Deko	Odenwälder Töpferwelt	Erbach	80 €

- Welche Attribute sind von Schlüsselkandidaten voll funktional abhängig?
- Welche Attribute sind von Schlüsselteilen voll funktional abhängig?



Vorgehen zur Auflösung zur 2NF

Für jeden(!) Teilschlüssel für den voll funktional abhängige Attribute existieren:

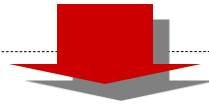
1. neue Relation anlegen, welche den Teilschlüssel und die von diesem voll funktional abhängigen Attribute enthält
2. abhängige Attribute aus der Originalrelation entfernen
3. Teilschlüssel in Originalrelation als Schlüssel und außerdem als Fremdschlüssel auf neue Relation deklarieren



Auflösung zur 2NF

Toeferprodukt_Markt

<u>Prod-Nr</u>	Produktart	Funktion	<u>Verkaufsmarkt</u>	Marktstandort	marktspez.Preis
11022	Tee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
10622	Kaffee-Service	Gebrauch	Internat. Tonmarkt	Strasbourg	200 €
20131	Schale	Deko	Rheinischer Tonmarkt	Mainz	80 €
20131	Schale	Deko	Odenwälder Töpferwelt	Erbach	50 €
20131	Schale	Deko	Internat. Tonmarkt	Strasbourg	120 €
40030	Krug	Deko	Internat. Tonmarkt	Strasbourg	100 €
40031	Krug	Deko	Odenwälder Töpferwelt	Erbach	80 €



Toeferprodukt

<u>Prod-Nr</u>	Produktart	Funktion
11022	Tee-Service	Gebrauch
10622	Kaffee-Service	Gebrauch
20131	Schale	Deko
40030	Krug	Deko
40031	Krug	Deko

Toeferprodukt_Markt_Neu

<u>Prod-Nr</u>	<u>Verkaufsmarkt</u>	marktspez.Preis
11022	Internat. Tonmarkt	200 €
10622	Internat. Tonmarkt	200 €
20131	Rheinischer Töpfermarkt	80 €
20131	Odenwälder Töpferwelt	50 €
20131	Internat. Tonmarkt	120 €
40030	Internat. Tonmarkt	100 €
40031	Odenwälder Töpferwelt	80 €

Toefermarkt

<u>Verkaufsmarkt</u>	Marktstandort
Internat. Tonmarkt	Strasbourg
Rheinischer Töpfermarkt	Mainz
Odenwälder Töpferwelt	Erbach



Beispiel zur 2NF

PRÜFUNG (MATRNR, LVNR, LVBEZEICHNUNG, DOZENT, RAUM, NOTE)

- *Annahme: jede Vorlesung wird von genau einem Dozenten gehalten und findet in genau einem Raum statt.*
- Ist das Schema in 2NF? (Welche Attribute sind von Schlüsselteilen voll funktional abhängig?)
- Ggf. Auflösung zur 2NF



Dritte Normalform (3NF)

Eine Relationenschema ist in **dritter Normalform**, wenn

- es in zweiter Normalform ist und
- kein Nichtschlüsselattribut *transitiv* von einem Schlüssel abhängt.

- Eine Attributmenge C hängt von einer Attributmenge A **transitiv** ab, wenn es eine Attributmenge B gibt, so dass gilt: $A \rightarrow B$ und $B \rightarrow C$.
- Anders ausgedrückt: ein Nichtschlüsselattribut darf **nicht** voll funktional abhängig von anderen Nichtschlüsselattributen sein, sondern nur von Schlüsselattributen
- Indirekte Abhängigkeiten vom Schlüssel über Nichtschlüsselattribute bedeutet i.a. dass der gleiche Fakt mehrfach gespeichert wird, d.h. Redundanz und daraus folgend Anomalien.
- Beispiel: BESTELLUNG (B-NR, B-DATUM, LIEFERANT-NR, LIEFERANT-NAME, LIEFERANT-ADRESSE)
 - $\{B-NR\} \rightarrow \{B-DATUM\}$
 - $\{B-NR\} \rightarrow \{LIEFERANT-NR\}$
 - $\{LIEFERANT-NR\} \rightarrow \{LIEFERANT-NAME\}$
 - $\{LIEFERANT-NR\} \rightarrow \{LIEFERANT-ADRESSE\}$



Vorgehen zur Auflösung zur 3NF

Für jede(!) transitiv abhängige Attributmenge C :

1. neue Relation anlegen, welche die transitiv abhängige Attributmenge C und die Attributmenge B (mit $A \rightarrow B$ und $B \rightarrow C$) enthält (B wird Schlüssel in neuer Relation)
2. transitiv abhängige Attribute aus der Originalrelation entfernen
3. Attributmenge B in der Originalrelation als Fremdschlüssel auf die neue Relation deklarieren

Beispiel:

BESTELLUNG (B-NR, B-DATUM, LIEFERANT-NR, LIEFERANT-NAME, LIEFERANT-ADRESSE)



LIEFERANT (LIEFERANT-NR, LIEFERANT-NAME, LIEFERANT-ADRESSE)

BESTELLUNG_NEU (B-NR, B-DATUM, LIEFERANT-NR \rightarrow LIEFERANT)



Zusammenfassung

- Im Relationenmodell ist die 1NF *immer erforderlich*. Ein Umwandlung in 2NF und 3NF ist *immer möglich*.
 - Normalisierung gemäß der 2NF und 3NF unterstützt die Gewährleistung referentieller Integrität insbesondere bei schreibenden, d.h. verändernden Zugriffen – für lesende Zugriffe ist sie nicht zwingend notwendig.
 - Aber auch: Normalisierung – insbesondere 3NF – führt u.U. zu reduzierter Laufzeit-Performance (Informationen müssen bei jedem Zugriff u.U. aus mehreren Tabellen zugesammengefügt werden)
- ⇒ In der Praxis wird – zur Performance-Optimierung – teilweise auf 3NF verzichtet („Denormalisierung“)

Empfehlungen

- Bereits bei der Entity-Relationship-Modellierung „normalisiert“ denken!
- Zuerst normalisieren und nur bei Performance-Problemen die nachweislich auf die NF zurückzuführen sind, u.U. „denormalisieren“.



Kapitel 3: Relationenmodell

- ✓ Grundlagen des Relationenmodell
- ✓ Abbildung des Entity-Relationship-Modells auf das Relationenmodell
- ✓ Normalformen



Vorlesung Datenbanken 1

