

Kapitel 3

Relationenmodell



3.1 Einleitung

3.2 Mathematische Relationen

3.3 Relationen in DBS

3.4 Schlüssel

3.5 Vom ER-Modell zum relationalen Modell

3.6 Relationenalgebra



3.1 Einleitung



Grundlage der meisten derzeitigen Datenbanken
vorgestellt von E.F. Codd 1970

E.F. Codd, „A Relational Model for Large Shared Data Banks“, CACM, Vol. 13, No. 6, 1970

Ziel: Daten-Unabhängigkeit

in kommerziellen DBMS seit 1981

basiert auf einer Variante des mathematischen Konzepts der **Relation**

Relationen können auf einfache Weise als **Tabellen** interpretiert werden
rein **wertbasierte** Verweise (keine Pointer)

Ein einfaches Beispiel für Bankkunden-Tabelle

Kunde-Nr	Kunde-Name	Adresse	Konto-Nummer	Filiale	Saldo
12345	Huber	Wien	G-2317	Wien	1.050,00
12346	Meier	Linz	G-2555	Wels	20.150,00
22335	Müller	Salzburg	K-1123	Wals	-40.000,00



D_1, D_2, \dots, D_n seien **n Mengen**

Das **kartesische Produkt** $D_1 \times D_2 \times \dots \times D_n$ ist die Menge aller **n-Tupel** (d_1, d_2, \dots, d_n) , sodass $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$

Eine **mathematische Relation** auf D_1, D_2, \dots, D_n ist eine Teilmenge des kartesischen Produktes $D_1 \times D_2 \times \dots \times D_n$

D_1, D_2, \dots, D_n sind **Wertebereiche** (Domänen) der Relation

n ist der **Grad der Relation**

Anzahl der Tupel wird **Kardinalität der Relation** genannt

Mengeneigenschaft, d.h. kein Tupel darf doppelt vorkommen

Beispiel

$\text{konto} \subseteq \text{integer} \times \text{string} \times \text{string} \times \text{string} \times \text{string} \times \text{number}$

$\{ (12345, \text{„Huber“}, \text{„Wien“}, \text{„G-2317“}, \text{„Wien“}, 1.050,00), \dots \}$

Attribute positional

12345	Huber	Wien	G-2317	Wien	1.050,00
12346	Meier	Linz	G-2555	Wels	20.150,00
22335	Müller	Salzburg	K-1123	Wals	-40.000,00



Jeder Domain erhält eindeutigen Attributnamen

beschreibt die „Rolle“ des Domains

Attributwerte sind **atomar**, d.h. sind nicht teilbar

z.B. der Wert eines Attributs kann eine „Konto-Nr“ sein, aber darf keine Menge von „Konto-Nr“n sein

In Tabellen-Darstellung sind die Attributnamen die Spaltenüberschriften

Bsp.:

Kunde-Nr	Kunde-Name	Adresse	Konto-Nummer	Filiale	Saldo
12345	Huber	Wien	G-2317	Wien	1.050,00
12346	Meier	Linz	G-2555	Wels	20.150,00
22335	Müller	Salzburg	K-1123	Wals	-40.000,00



Relationenschema

Name einer Relation R mit einer Menge von Attributen A_1, \dots, A_n

$$R(A_1, \dots, A_n)$$

Datenbankschema

Menge von Relationenschemata mit unterschiedlichen Namen

$$R = \{ R_1(X_1), R_2(X_2), \dots, R_n(X_n) \}$$

Relation (Instanz) r mit einem Schema $R(X)$

Menge r von Tupeln gemäß X

Datenbank (Instanz)

Zu einem Datenbankschema $R = \{ R_1(X_1), R_2(X_2), \dots, R_n(X_n) \}$ repräsentiert die Menge von Relationen $r = \{ r_1, \dots, r_n \}$ mit r_i Relation auf R_i die Datenbank



DB-Schema „Unternehmen“:

Mitarbeiter (MNr, MName, GebDatum, AbtNr)

Abteilung (AbtNr, Bez, Ort, Leiter)

Kurs (KursBez, Leiter, Anzahl_Tage)

Zertifikat (MNr, KursBez, Punkte, Datum)

Beispielrelation: Mitarbeiter

MNr	MName	GebDatum	AbtNr
4541	Huber	1.12.1965	16
4545	Berger	3.5.1952	16
4567	Dorfer	1.2.1973	23
4599	Maier	11.5.1970	23
4587	Müller	13.8.1969	12

Vorteile

Wertbasiert

Jede Information in relationalen DBen wird durch **Werte** ausgedrückt
Referenzen (Verweise) sind ebenfalls nur Werte

logisches Modell unabhängig von physischer Struktur

enthält nur Information, die aus Anwendersicht relevant ist, **keine Implementierungsdetails**

Leichte **Transferierbarkeit** von Daten zwischen Systemen

keine Pointer!



Bedeutung von **Nullwerten**

unbekannter Wert

es gibt einen Wert, aber er ist nicht bekannt, Bsp.: Durchwahl neuer Mitarbeiter

nicht existierender Wert

Attribut ist nicht anwendbar, Bsp.: Durchwahl freier Mitarbeiter

keine Information

unbekannt, ob Wert existiert oder nicht, Bsp.: Durchwahl eines Mitarbeiters, bei dem unklar ist, ob er ein freier oder fest angestellter Mitarbeiter ist

In DBMS: implizit „keine Information“

MNr	MName	Durchwahl
4541	Huber	25
4545	Berger	
4567	Dorfer	50
4599	Maier	
4587	Müller	

Interpretation im Beispiel:

Jeder feste Mitarbeiter hat eine Telefondurchwahl

- Huber hat die Durchwahl 25 und Dorfer die 50
- Maier ist freier Mitarbeiter (und hat keinen Schreibtisch im Haus)
- Die Durchwahl von Berger ist nicht bekannt
- Müller wurde neu eingestellt - wir wissen nicht, ob sie schon einen Telefonanschluss erhalten hat



„Codierung“ unvollständiger Information

schlecht: nicht verwendete Werte des Wertbereichs

Bsp.: 0, 99, blank, leere Zeichenkette

es muss keine solchen Werte geben

„nicht verwendete“ Werte könnten Bedeutung bekommen

Probleme bei Funktionen, z.B. Durchschnittsalter, wenn der Wert 0 als Platzhalter für unbekannte Werte verwendet wird.

Nullwert

spezieller Wert (kein Wert des Wertebereiches), der ausdrückt, dass kein Wert angegeben ist

Wird in Datenbanksystemen üblicherweise mit **NULL** angegeben

Beispiele in SQL

```
SET Durchwahl = NULL
```

```
WHERE Durchwahl IS NOT NULL
```



Eine Relation ist eine Menge von Tupeln, d.h. Zeilen müssen verschieden sein („**Mengeneigenschaft**“)

Aber: In einer allgemeinen Tabelle könnten 2 Zeilen gleich sein - Probleme beim Zugriff auf eine bestimmte Zeile

Wenn keine Zeile mehrfach vorhanden sein kann - dann enthält die Tabelle einen **Schlüssel**

Ein Teilmenge von Attributen des Relationenschemas heißt **Oberschlüssel**, wenn sie die einzelnen Tupeln in der Relation eindeutig identifiziert

Ein **Oberschlüssel** (Superkey) einer Relation $r(X)$ ist eine Teilmenge K von X , sodass für je zwei verschiedene Tupel u und v aus $r(X)$ immer gilt $u(K) \neq v(K)$



Wir betrachten ein erweitertes Beispiel

Mitarbeiter (MNr, MName, SV-Nummer, GebDatum, AbtNr)

Beispiele für mögliche
Oberschlüssel sind:
(MNr, MName, GebDatum),
(SV-Nummer)

MNr	MName	SV-Nummer	GebDatum	AbtNr
4541	Huber	12340112651	1.12.1965	16
4545	Berger	23451308692	13.8.1969	16
4567	Dorfer	34560102733	1.2.1973	23
4599	Maier	45671105704	11.5.1970	23
4587	Müller	56781308695	13.8.1969	12
4588	Berger	67890405526	4.5.1952	23

Ein **Schlüsselkandidat** ist ein minimaler (kleinster)
Oberschlüssel

d.h. wenn ein Attribut gestrichen wird, dann ist der Schlüssel kein
Oberschlüssel mehr

Beispiele für Schlüsselkandidaten sind im obigen Beispiel
(MNr),
(SV-Nummer)



Ein Schlüsselkandidat kann aus **mehreren Attributen** bestehen

Zeugnis (MNr, KursBez, Datum, Punkte)

Ein Mitarbeiter kann mehrere unterschiedliche Zeugnisse haben, die auch am selben Tag ausgestellt wurden

Zu einem Kurs können an einem Tag mehrere Zeugnisse ausgestellt werden

Zeugnis

MNr	KursBez	Datum	Punkte
4541	DBS	1.12.2005	26
4541	OOProg	1.12.2005	16
4567	OOProg	1.12.2005	23
4541	ITEng	11.5.2000	23
4587	OOProg	1.12.2005	12

Der Schlüssel ist daher
(MNr, KursBez, Datum)



Wir betrachten das vorhergehende Beispiel

Mitarbeiter (MNr, MName, SV-Nummer, GebDatum, AbtNr)

In einer Relation kann es mehrere Schlüsselkandidaten geben

Bsp.: (SVNr), (MNr)

Der **Schlüssel (Primärschlüssel)** ist ein (beliebig) ausgewählter Schlüsselkandidat mit dem ein Tupel referenziert werden soll

Primärschlüssel werden im Relationenschema durch **Unterstreichen** gekennzeichnet.

Beispiel

MNr sei Primärschlüssel

Mitarbeiter (MNr, MName, SV-Nummer, GebDatum, AbtNr)

Für einen guten Primärschlüssel gilt

Wert immer bekannt

Wert ändert sich nie

... deshalb: SVNr, MNr, KontoNr,



Mit **Fremdschlüssel** bezeichnen wir das Attribut einer Relation, das in einer anderen Relation Schlüssel ist

Beispiel

Mitarbeiter (MNr, MName, SVNr, GebDatum, *AbtNr*)

Abteilung (AbtNr, Bez, Ort, *Leiter*)

Im Beispiel verweist Attribut *Leiter* in Abteilung auf Attribut MNr in Mitarbeiter, und Attribut *AbtNr* in Mitarbeiter auf Abteilung

Fremdschlüssel realisieren Verweise auf andere Relationen

Fremdschlüssel ermöglichen die Integritätsbedingung der **referentiellen Integrität**, die sichert, dass zu Werten eines Fremdschlüssels in einer Relation, die auf Tupel in einer anderen Relation verweisen, immer auch die entsprechenden Tupel vorhanden sind

Fremdschlüssel werden im Relationenschema üblw. **kursiv** dargestellt

Manchmal findet man die „**Diamant**“ **Schreibweise** (siehe Bsp. oben):

Abteilung.*Leiter* ◇ Mitarbeiter, Mitarbeiter.*AbtNr* ◇ Abteilung



ER-Diagramme verwendet man für den konzeptuellen Entwurf

Das relationale Modell ist **semantisch ärmer**

Nicht alles was im ER-Modell ausgedrückt werden kann, kann auch im relationalen Modell ausgedrückt werden

Integritätsbedingungen im Datenmodell

Vorgaben für Programmierer und Benutzer

Abbildungsmethodik notwendig

Schritte:

1. Vorbereiten des konzeptuellen Modells
2. Abbilden auf Relationen

Vorgangsweise ist analog für die Abbildung von UML-Diagrammen auf relationales Modell



1. Vorbereiten

1. Eliminieren von Mehrfach-Attributen
2. Eliminieren von komplexen Attributen

2. Abbilden auf Relationen

1. Entität wird Relation
2. $m:n$ Assoziation wird Relationen
3. $1:m$ Assoziation wird (kann) integriert (werden)
4. $1:1$ Assoziation wird (kann) integriert (werden)
5. Assoziationen höheren Grades werden Relation

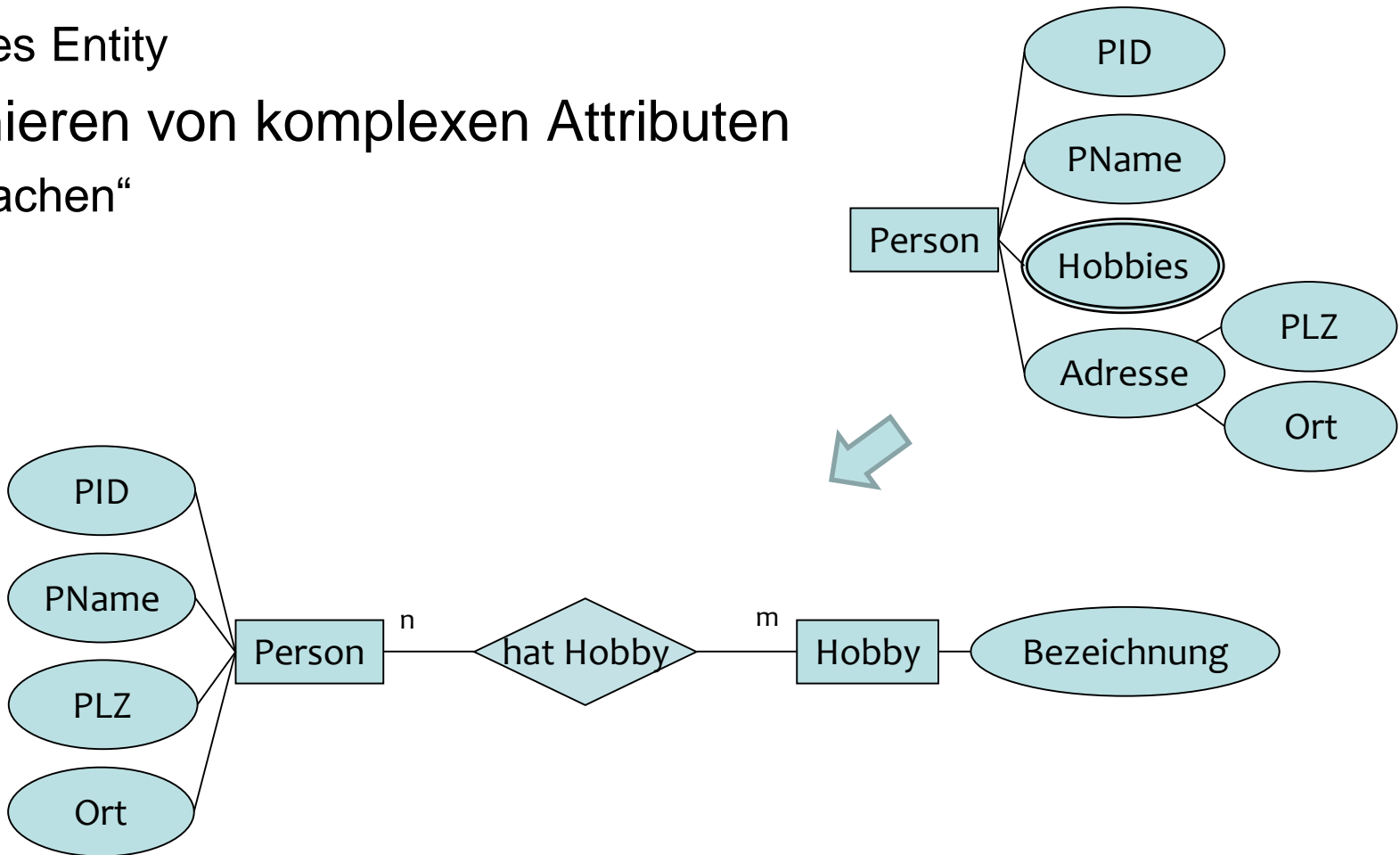


1. Eliminieren von Mehrfach-Attributen

eigenes Entity

2. Eliminieren von komplexen Attributen

„ausflachen“



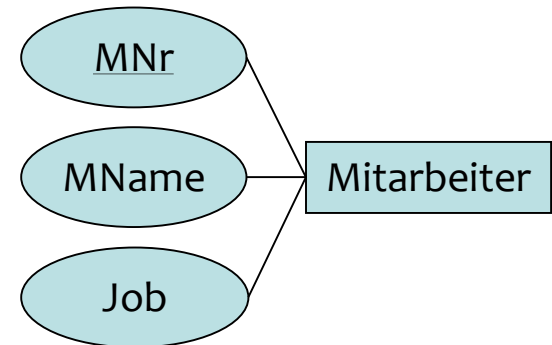
1. Abbildungsregel: Entity wird Relation



Jedes Entity wird auf eine eigene
Relation abgebildet

Name des Entity ist Name der
Relation

Attribute des Entity sind Attribute der
Relation



Mitarbeiter(MNr, MName, Job)



1a. Abbildungsregel: Schwache Entitäten



Jedes schwache Entity wird auf eine eigene
Relation abgebildet

Name des Entity ist Name der Relation

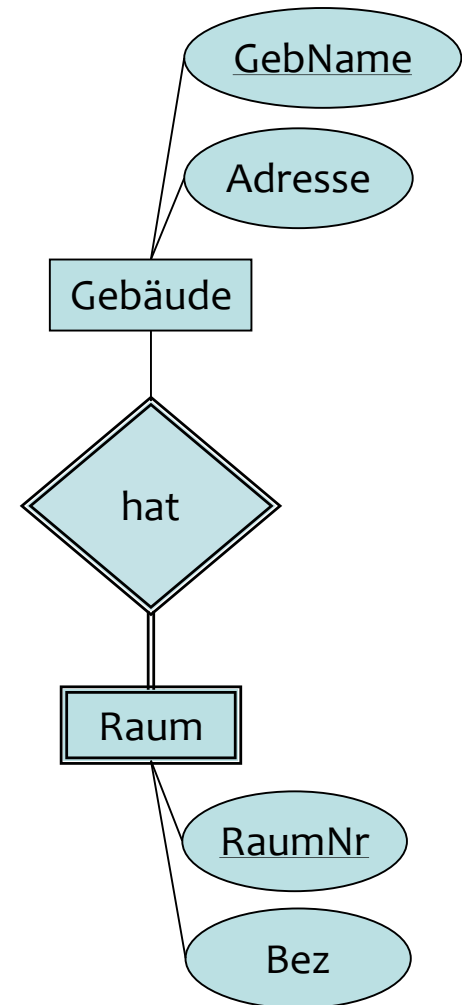
Attribute des Entity sind Attribute der Relation

Füge den Primärschlüssel des bestimmenden
starken Entity als Fremdschlüssel in die Relation
ein

Der Primärschlüssel der Relation ergibt sich aus
der Kombination des Primärschlüssels des
bestimmenden Entity und des partiellen
Schlüssels des schwachen Entity

Gebäude(GebName, Adresse)

Raum(GebName, RaumNr, Bez)



2. Abbildungsregel: m:n Assoziation



Jede m:n Assoziation wird auf eine eigene
Relation abgebildet

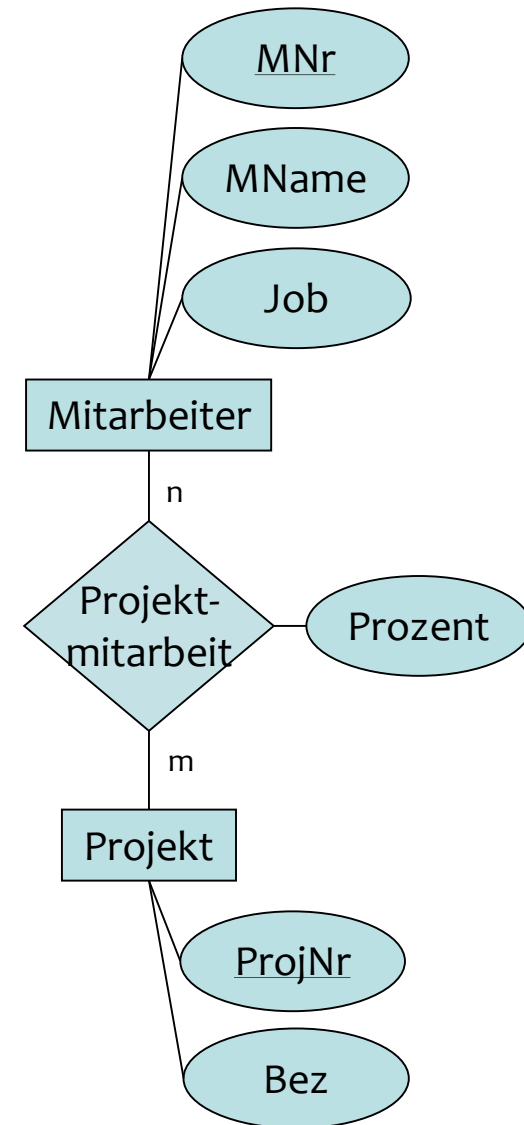
Name der Relation ist der Name der Assoziation

Attribute der Relation sind

Schlüsselattribute der beteiligten Entities
(bei rekursiven Beziehungen erweitert
um die Rollenbezeichnung)

Attribute der Assoziation

Projektmitarbeit (MNr, ProjNr, Prozent)



3. Abbildungsregel: 1:n Assoziation

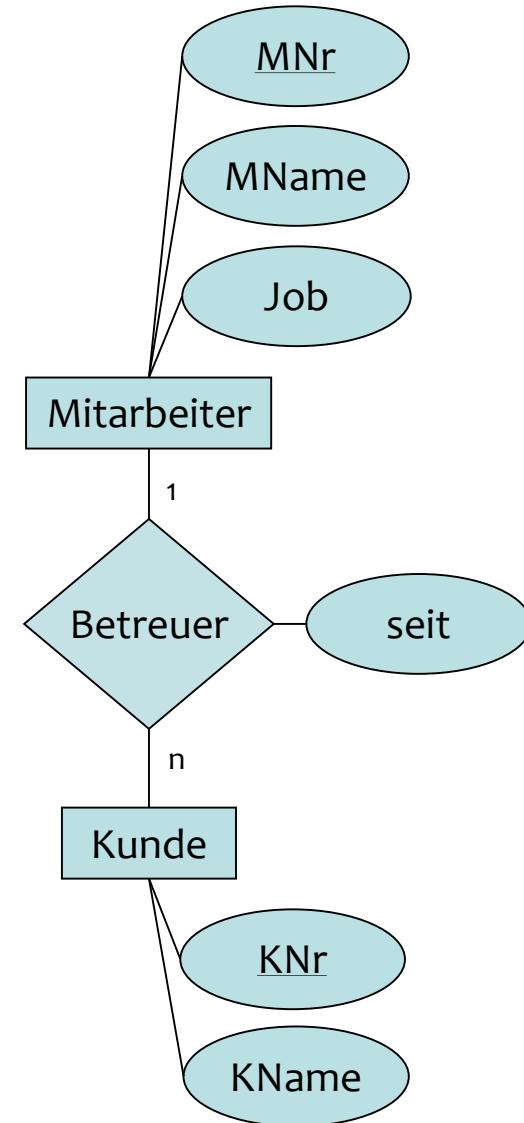


1:n Assoziation

Hier muss keine eigene Relation gebildet werden

Die Relation des mehrfach vorkommenden Entity wird um den Primärschlüssel der (Relation des) einfach vorkommenden Entity und der Assoziations-Attribute erweitert.

Kunde(KNr, KName, *Betreuer-MNr*, seit)



4. Abbildungsregel: 1:1 Assoziation



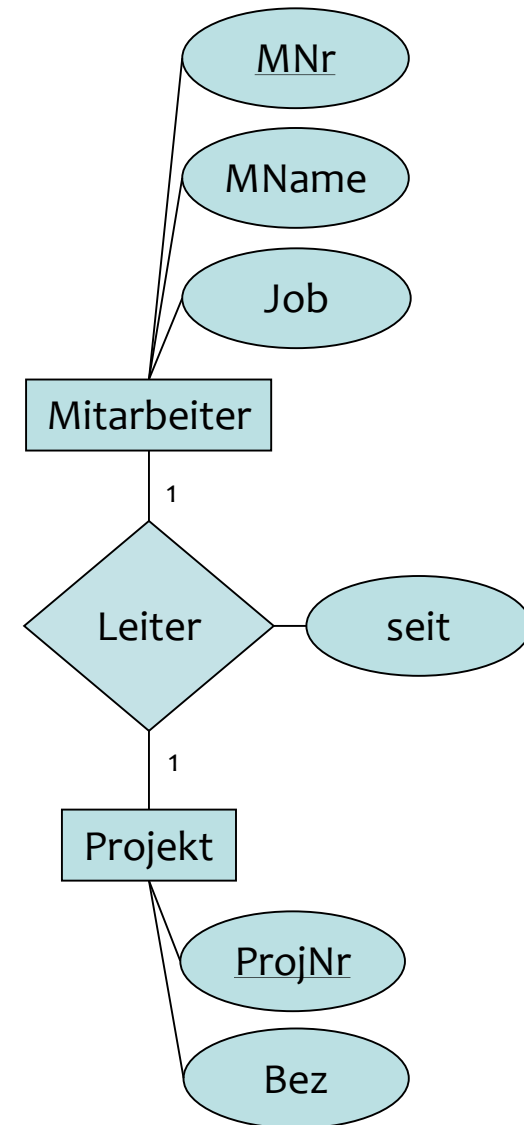
1:1 Assoziationen

Ebenfalls keine eigene Relation
eine der beiden Relationen wird um den
Primärschlüssel und um die
Assoziationsattribute der anderen erweitert.

Projekt(ProjNr, Bez, *Leiter-MNr*, seit)

ODER

Mitarbeiter(MNr, Mname, Job, *ProjNr*, seit)



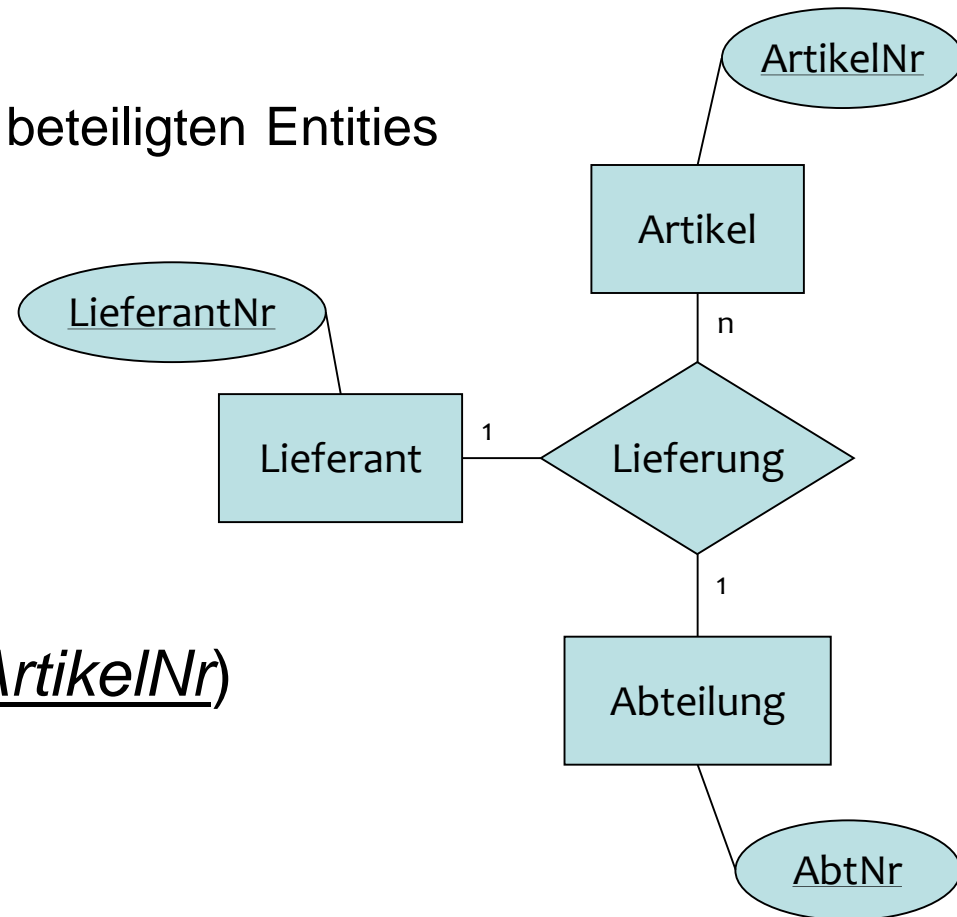
5. Abbildungsregel: n-äre Assoziationen



Assoziationen höheren Grades

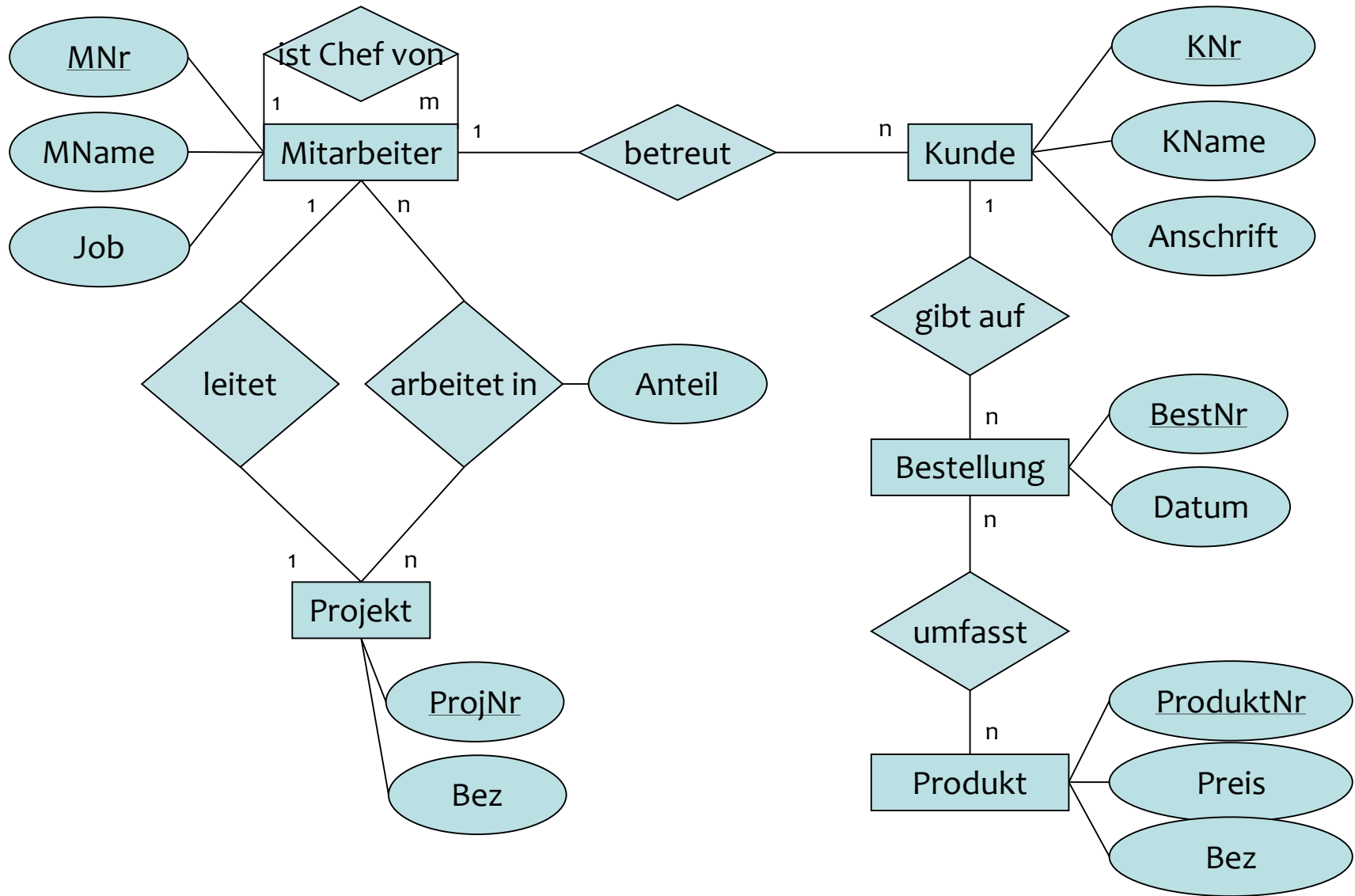
eigene Relation

besteht aus Primärschlüsseln der beteiligten Entities
und den Beziehungsattributen



Lieferung (LieferantNr, AbtNr, ArtikelNr)





Mitarbeiter(MNr, MName, Job, *Chef-MNr*)

Projekt(ProjNr, Bez, *Leiter-MNr*)

Kunde(KNr, KName, Anschrift, *Betreuer-MNr*)

Bestellung(BestNr, Datum, *Besteller-KNr*)

Produkt(ProdNr, Bez, Preis)

Projektarbeit(MNr, ProjNr, Anteil)

Bestellumfang(BestNr, ProdNr)



Die **Relationenalgebra** ist eine formale Sprache, mit der sich Anfragen über einem relationalen Schema formulieren lassen
Relationen können miteinander verknüpft oder reduziert und spezifische Informationen daraus hergeleitet werden

Es werden Operationen auf Relationen definiert

Operationen sind abgeschlossen, d.h. Ergebnis ist wieder eine Relation

- Mengenoperationen
 - Vereinigung \cup , Differenz $-$, Durchschnitt \cap
- Selektion σ
- Projektion π
- Verbund \bowtie
- Umbenennung ρ

Weitere Operationen (hier nicht behandelt)

- Semi-Verbund \ltimes
- Division \div
- ...



Vereinigung \cup , **Durchschnitt** \cap und **Differenz** – von zwei Relationen, definiert über dem gleichen Relationenschema

Beispiele:

r	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>a1</td><td>b1</td><td>c1</td></tr><tr><td>a1</td><td>b2</td><td>c1</td></tr><tr><td>a2</td><td>b1</td><td>c2</td></tr></table>	A	B	C	a1	b1	c1	a1	b2	c1	a2	b1	c2
A	B	C											
a1	b1	c1											
a1	b2	c1											
a2	b1	c2											

S	A	B	C
	a1	b2	c1
	a2	b2	c1
	a2	b2	c2

$r \cup s =$	A	B	C
	a1	b1	c1
	a1	b2	c1
	a2	b1	c2
	a2	b2	c1
	a2	b2	c2

$\mathbf{r} \cap \mathbf{s} =$	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>a1</td><td>b2</td><td>c1</td></tr></table>	A	B	C	a1	b2	c1
A	B	C					
a1	b2	c1					

r - s =	A	B	C
	a1	b1	c1
	a2	b1	c2



Die **einfache Selektion** σ definiert eine Auswahl einer Teilmenge von Tupeln aufgrund einer Gleichheits-Bedingung

$$\sigma_{A=a}(r) = \{ t \in r \mid t(A) = a \}$$

Beispiele: r

A	B	C
a1	b1	c1
a1	b2	c1
a2	b1	c2

$$\sigma_{C=c1}(r) =$$

A	B	C
a1	b1	c1
a1	b2	c1

Eigenschaften

Relationenschema bleibt gleich

kommutativ: $\sigma_{A1=a}(\sigma_{A2=b}(r)) = \sigma_{A2=b}(\sigma_{A1=a}(r))$

Beispiel: $\sigma_{\text{Name}=\text{„Huber“}}$ (Mitarbeiter)
alle Mitarbeiter mit Namen „Huber“



Die **Theta(θ) - Selektion** erlaubt beliebige Vergleichsoperatoren als Selektions-Bedingung

Sei Θ eine Menge von Vergleichsoperatoren auf
 $D_i = \text{Domain}(A) = \text{Domain}(B)$ und θ ein Operator aus dieser Menge

$$\sigma_{A \theta a}(r) = \{t \in r \mid t(A) \theta a\}$$

$$\sigma_{A \theta B}(r) = \{t \in r \mid t(A) \theta t(B)\}$$

Beispiel: $\sigma_{MNr > 100}$ (Mitarbeiter)
alle Mitarbeiter mit MNr größer 100



Bei der **Projektion** $\pi(r)$ wird eine Auswahl von Spalten der Argumentrelation r extrahiert

$$\text{Sei } X \subseteq R, \pi_X(r) = \{t(X) \mid t \in R\}$$

Beispiel:

r

A	B	C
a1	b1	c1
a1	b1	c2
a2	b1	c1

$\pi_{A,B}(r) =$

A	B
a1	b1
a2	b1

Eigenschaften

- Vertauschbar mit Selektion, wenn alle Attribute, auf denen selektiert wird, in den projizierten Attributen enthalten sind,
d.h. $\sigma_{MNr>10} \pi_{MNr, MName} \text{Mitarbeiter} = \pi_{MNr, MName} \sigma_{MNr>10} \text{Mitarbeiter}$,
- $|\pi_X(r)| \leq |r|$

Beispiel: $\pi_{MName}(\text{Mitarbeiter})$

Die Namen aller Mitarbeiter



Das **kartesische Produkt (Kreuzprodukt)** $r \times s$ zweier Relationen r und s enthält alle möglichen Paare von Tupeln aus r und s

Beispiel:

r

A	B
a1	b1
a1	b2
a2	b3

s

C	D
c1	d1
c2	d1

$r \times s =$

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d1
a1	b2	c1	d1
a1	b2	c2	d1
a2	b3	c1	d1
a2	b3	c2	d1



Der **natürliche Verbund** $r \bowtie s$ wird durch das Kreuzprodukt gebildet, aus dem nur diejenigen Tupel selektiert werden, deren Attributwert für gleichbenannte Attribute der beiden Relationen gleich sind, d.h.

$t_r \in r$ und $t_s \in s$, falls t_r und t_s denselben Wert für jedes Attribut $R \cap S$, dann ist $t_r \times t_s = t$ Tupel des Ergebnisses

Beispiel:

r	s	$r \bowtie s =$
A a1 a1 a1 a1	B b1 b1 b2 b1	C c1 c2 c1 c3
C c1 c1 c2	D d1 d2 d1	A a1 a1 a1 a1
C c1 c1 c2	D d1 d2 d1	B b1 b1 b2 b2
C c1 c1 c2	D d1 d2 d1	C c1 c2 c1 c1
C c1 c1 c2	D d1 d2 d1	D d1 d2 d1 d2

Beispiel: Student (MatrNr, SName, StKZ)

Str (StKZ, Bezeichnung, Semester)

Student \bowtie Str = (MatrNr, SName, StKZ, Bezeichnung, Semester)



kommutativ: $r \bowtie s = s \bowtie r$

distributiv mit Selektion: $\sigma_{A=a} (r \bowtie s) = (\sigma_{A=a} (r) \bowtie \sigma_{A=a} (s))$

assoziativ: $r \bowtie (s \bowtie q) = (r \bowtie s) \bowtie q$

$$|r \times s| = |r| \cdot |s|$$

$$|r \bowtie s| \leq |r| \cdot |s|$$

Falls $R \cap S = \{\}$, dann ist $r \bowtie s$ das kartesische Produkt von r und s : $r \times s$



Der **Allgemeine Verbund** $r \bowtie_{\theta} s$ ($r [A \theta B] s$) erlaubt die Angabe eines beliebigen Verbund-Prädikats θ

Seien $r (R)$ und $s (S)$ Relationen mit $R \cap S = \{\}$.

Sei $A \in R$, $B \in S$ und Θ eine Menge von Vergleichsoperatoren auf $\text{Domain}(A) \times \text{Domain}(B)$,

Sei θ aus Θ

$$\begin{aligned} r [A \theta B] s &= q (RS) = \{t \mid \exists tr \in r, \exists ts \in s \text{ mit} \\ &\quad t (R) = tr, t (S) = ts \text{ und } t (A) \theta t (B)\} \\ &= \sigma_{A \theta B} (r \times s) \end{aligned}$$

Beispiel: r

A	B	C
a1	b1	1
a1	b1	2
a1	b2	1
a1	b1	4

s

D	E
1	e1
1	e2
2	e1
3	e1

$r [C < D] s =$

A	B	C	D	E
a1	b1	1	2	e1
a1	b1	1	3	e1
a1	b1	2	3	e1
a1	b2	1	2	e1
a1	b2	1	3	e1



Durch diese Operation können Attribute und Relationen umbenannt werden. Diese Operation ist wichtig, um Joins von unterschiedlichen benannten Relationen zu ermöglichen, kartesische Produkte zu ermöglichen, wo es gleiche Attributnamen gibt, insbesondere auch mit der gleichen Relation Mengenoperationen zwischen Relationen mit unterschiedlichen Attributen zu ermöglichen.

Notation

Falls der Ausdruck aus n Attributen besteht, liefert $\rho_{A_1=B_1, A_2=B_2, \dots, A_n=B_n}(E)$ das Ergebnis von E mit den Attributen A_1, \dots, A_n auf B_1, \dots, B_n umbenannt

Beispiel:

r	A	B	$\rho_{A=C, B=D}(\sigma_{A=a1} \ r) =$	C	D
	a1	b1		a1	b1
	a2	b1			



Mitarbeiter(MNr, MName, Job, *Chef-MNr*)

Projekt(ProjNr, Bez, *Leiter-MNr*)

Kunde(KNr, KName, Anschrift, *Betreuer-MNr*)

Bestellung(BestNr, Datum, *Besteller-KNr*)

Produkt(ProdNr, Bez, Preis)

Projektarbeit(MNr, ProjNr, Anteil)

Bestellumfang(BestNr, ProdNr)

1. „Alle Produkte mit einem Preis größer 100

$\sigma_{\text{Preis} > 100}$ (Produkt)

2. „Die Bezeichnungen aller Projekte“

π_{Bez} (Projekt)

3. „Projektbezeichnungen und alle daran beteiligten Mitarbeiternamen“

$\pi_{\text{Bez, MName}}$ (Mitarbeiter \bowtie Projektarbeit \bowtie Projekt)

4. „Alle Chefs mit Ihren Mitarbeitern“

$\pi_{\text{MN1, MN2}} (\rho_{\text{MName=MN1, MNr=M}} \text{ (Mitarbeiter)} [M=C] \rho_{\text{MName=MN2, Chef-MNr=C}} \text{ (Mitarbeiter)})$

