Folien zur Vorlesung Datenbanken

Kapitel 5-Teil B

Grundlagen des relationalen Datenmodells (Wiederholung und detailliertere Darstellung)

Fachhochschule Wedel

Prof. Dr. Ulrich Hoffmann

basierend auf den Lehrmaterialien von
Prof. Dr. Hans–Detlef Gerhardt

5.3 Grundlagen des relationalen Datenmodells (Wiederholung und detailliertere Darstellung)

5.3.1 Relationen

Beispiel:

„Datenmodell“ Karteikarte (BUCHBESTAND)

INVNR	....	AUTOR	TITEL	VERLAG	...
2		Meier	Inf2	Springer	
3		Kron	DBS	IWT	
4		Liskin	ORACLE	Hill	

X = $\{A_1, \ldots, A_m\}$

dom(X) = $\bigcup_{A \in X} \mathrm{dom}(A)$

Tup (X) Menge aller Tupel über X

Relation r über X: endliche Menge von Tupeln über X, d.h. $\mathbf{r} \subseteq \mathbf{Tup(X)}$

Rel(X) Menge aller Relationen über X

Relation über X: in Form einer Tabelle darstellbar, sofern eine Reihenfolge der Attribut von X festgelegt ist.

ERM	RDM
Entity	Tupel einer Relation
Entity-Set	Relation

Problem:
Einem Attribut kann aktuell kein Wert zugeordnet werden.

Ein **partielles** Tupel über X ist eine Abbildung

$$\mu : \mathbf{X} \rightarrow \mathbf{dom(X)} \cup \mathbf{NULL},$$

für die für mindestens ein $A \in X$ gilt

$$\mu(\mathbf{A}) \in \mathbf{dom(A)} \quad \textbf{und} \quad \mu(\mathbf{A}) = \mathbf{NULL}.$$

Relation r über X heißt **partiell**, falls sie **partielle** Tupel enthält.

5.3.2 Relationenschemas und Datenabhängigkeiten

Rel(X)
Nur die Relationen von Rel(X) interessieren, die **bestimmten semantischen Bedingungen** genügen, d. h. einen aktuellen „Ausschnitt der realen Welt" repräsentieren.

Beispiel:
X = {INVNR, AUTOR, TITEL, VERLAG}
Zwei verschiedene Relationen r_1, $r_2 \in$ Rel(X):

	INVNR	AUTOR	TITEL	VERLAG
r_1	1	Meier	Inf1	Springer
	2	Meier	Inf2	Springer
	3	Kron	DBS	IWT
	4	Liskin	ORACLE	Hill
r_2	1	Meier	Inf1	Springer
	2	Meier	DBS	Springer
	1	Kron	DBS	IWT
	2	Liskin	ORACLE	Hill

r_1 sinnvolle Relation über X

r_2 nicht sinnvoll, weil

- Inventarnummer mehrfach vergeben
- gleiche Titel für verschiedene Bücher (nicht zulässig)

Diese Beobachtungen semantischer Natur werden als

Datenabhängigkeiten bezeichnet.

Datenabhängigkeiten liefern Aussagen, welche Relationen

$r \in Rel(X)$ als **„sinnvoll“** oder **„gültig“** anzusehen sind und daher in einer DB „erlaubt“ sein sollen.

Statische Datenabhängigkeit:

jede „aktuell vorliegende“ Relation $r \in Rel(X)$ erfüllt sie.

Dynamische Datenabhängigkeit:

betrifft Übergang von einer Relation
$r \in Rel(X)$ auf eine andere $r' \in Rel(X)$

Statische Abhängigkeiten (intrarelationale Abhängigkeiten):

Definition: Abbildung σ

σ **: Rel(X) $\rightarrow$ { 0, 1} mit**

{0, 1} als Menge der Boolschen Wahrheitswerte.

Falls σ **(r) = 1** für r $\in$ Rel(X), so sagt man
r erfüllt Abhängigkeit σ oder
σ **gilt in r.**

Definition: Abbildung $\sum_x$

$\sum_x$**: Rel(X) $\rightarrow$ { 0, 1}**
mit

$$\textstyle\sum_x(r) := \bigwedge_{\sigma \in \sum_x} \sigma(r).$$

r $\in$ Rel(X) erfüllt eine Menge **intrarelationaler Abhängigkeiten**, falls r **jede** Abhängigkeit $\sigma \in \sum_x$ erfüllt.

Primärschlüssel als Beispiel für eine **intrarelationale** Abhängigkeit:

Definition: Sei X eine Attributmenge, $K \subseteq X$.

1. **K heißt Schlüssel** für $r \in Rel(X)$, falls
   a) $(\forall \mu, \nu \in r) \quad \boldsymbol{\mu[K] = \nu[K]} \quad \Rightarrow \quad \boldsymbol{\mu = \nu}$
   b) für keine echte Teilmenge $\mathbf{K' \subset K}$ gilt (a).

2. Eine **Schlüsselabhängigkeit** $\boldsymbol{\sigma : K \to X}$
   bezeichnet folgende semantische Bedingung:
   Sei $r \in Rel(X)$:

$$(K \to X)(r) := \begin{cases} 1 \text{ falls K Schlüssel für r} \\ 0 \text{ sonst} \end{cases}$$

Die Auszeichnung einer Teilmenge K von X als Schlüssel bedeutet, dass eine Relation $r \in Rel(X)$ nur dann „gültiges Beispiel“ für X ist, falls sie $K \to X$ erfüllt, d. h.

$$\mathbf{(K \to X)\ (r) = 1} \text{ gilt.}$$

Eine Relation $r \in Rel(X)$ erfülle die **Schlüsselabhängigkeiten**

$\{K_1 \rightarrow X, K_2 \rightarrow X, \ldots, K_n \rightarrow X\}$ d.h. $\sigma_i = (K_i \rightarrow X)$.

Identifiziert man σ mit K, erhält man kurz die Menge von Schlüsselabhängigkeiten $\{K_1, K_2, \ldots, K_n\}$.

$n = 1 \Rightarrow$ Attribute des Schlüssels unterstreichen

$n > 1 \Rightarrow$ Primärschlüssel wählen und unterstreichen.

Schlüsselabhängigkeit schränkt die Menge aller Relationen Rel(X) ein auf die den **Schlüssel erfüllenden Relationen**.

Sei X **Relationenformat**

$\sum_x$ Menge von Abhängigkeiten über X.

Definition:

$$\textbf{Sat}(X, \textstyle\sum_x) := \{r \in Rel(X): \textstyle\sum_x (r) = 1\},$$

d.h. Menge aller Relationen $r \in Rel(X)$, welche $\sum_x$ erfüllen.

Definition: Ein Relationenschema hat die Form

$R = (X, \sum_x)$, wobei

R - Name des Relationenschemas
X - Relationenformat
$\sum_x$ - Menge von intrarelationalen Abhängigkeiten.

Relationenschema $R = (X, \sum_x)$ dient der **zeitinvarianten** Beschreibung der Menge **Sat $(X, \sum_x)$** aller Relationen über X, welche $\sum_x$ erfüllen.

Anstelle von $\sum_x$ wird oft nur $\sum$,
anstelle von **Sat $(X, \sum_x)$** wird oft nur **Sat(R)** geschrieben.

Betrachtet man nur die Abhängigkeit vom Primärschlüssel, so kann man festlegen:

Definition: Ein R-Schema hat die Form

R = (X, K), es umfasst

R - Name des R-Schema
X - Relationenformat (eine Attributmenge)
K - Primärschlüssel.

5.3.3 Relationale Datenbanken

Gegeben: Entity-Relationship-Diagramm

Gesucht: Relationen-Modell

Gegeben: Entity-Deklaration **E = (attr(E), K)**

Daraus ist das R-Schema

R = (X, K) abzuleiten.

R Name der Entity-Deklaration E wird übernommen.

Sei attr(E) = $\{A_1, ..., A_m\}$, daraus wird abgeleitet

X = $(A_1, ..., A_m)$

K wird als **Primärschlüssel** übernommen.

Gegeben: Relationship-Deklaration

R = (ent(R), attr(R)),

wobei

$ent(R) = (E_1 ..., E_k)$ und

$attr(R) = (B_1 ..., B_n)$.

$\mathbf{K_i}$ sei **Primärschlüssel** von $\mathbf{E_i}$, $i = 1(1)k$,

es gelte $\mathbf{K_i = (A_{i\,1}, ..., A_{i\,n_i}\ )}$.

Daraus wird das **R-Schema R** abgeleitet:

R = **(X, K)** mit

R Name der Relationship-Deklaration (wird übernommen)

X = $\{A_{11}, ..., A_{1n_1},..., A_{k1}, ..., A_{kn_k}, B_1 ..., B_n\}$

(die Namen aller in X vorkommenden Attribute seien verschieden)

Wie wird der Primärschlüssel für R ermittelt?

fhwedel UNIVERSITY OF APPLIED SCIENCES

Beispiel:

Gegeben ERM der Bibliothek

⇒ für die Entity-Deklarationen BUCH und LESER

die **R-Schemas**

BUCH = ({INVNR, AUTOR, TITEL, VERLAG}, {INVNR})

LESER = ({LNR, NAME, ADRESSE}, {LNR})

⇒ für die Relationship-Deklaration ENTLIEHEN

das **R-Schema**

ENTLIEHEN = ({INVNR, LNR, RÜCKGABEDATUM}, {INVNR}).

Ist R eine **1:n-Beziehung** zwischen E_1 und E_2, dann ist K_2 ein Primärschlüssel für das R-Schema R.

Ist R eine **n:m-Beziehung** zwischen E_1 und E_2, dann kann $K_1 \cup K_2$ als Primärschlüssel gewählt werden.

Ist R eine **1:1 Beziehung** zwischen E_1 und E_2, dann kann K_1 oder K_2 als Primärschlüssel für R gewählt werden.

Ist R **allgemeine mehrstellige Relationship-Deklaration**, dann könnte als Schlüsselabhängigkeit

$$\mathbf{K \rightarrow X \quad mit \quad K = X\text{-}(B_1, ..., B_n\}}$$

festgelegt werden.

Für n = 0 folgt daraus die triviale Schlüsselabhängigkeit $\mathbf{X \rightarrow X}$**.**

Ergebnis:

Menge R = $\{R_1, ..., R_k\}$ von Relationenschemas

ER-Modell	Relationen - Modell
Entity - Deklaration Relationship - Deklaration	R - schema
Entity-, Relationship-Set	Relation (Tabelle)
Entity, Relationship	Tupel (Zeile)

Konzeptionelle Datenmodellierung für ein betriebliches Informationssystem (9)

Übertragung des ER-Modells in das relationale Datenmodell: Ableitung von R-Schemas:

PERSON = ({PNR, NAME , ADR(PLZ,STADT, STRASSE), TEL, E-MAIL}, {PNR})

MITARBEITER = ({PNR, GEH_STUFE, BETRAG, ABT_NR, ABT_NAME, KRANKEN-KASSE, {P_BETRAG}, {KINDER (K_NAME, K_VORNAME, _GEB)}}, {PNR})

ZEITAK = ({PNR, L_FIRMA, H_LOHN, {VON_BIS}, H_SUMME}, {PNR})

MANAGER = ({PNR, A_GRAD, STUD_RI}, {PNR})

KB = ({PNR, KD_NAME}, {PNR})

MASCHINE = ({MNR, NAME, ANSCH_DATUM, NEUWERT, ZEITWERT}, {MNR})

ROHSTOFF = ({RNR, R_NAME, MENGE, PREIS}, {RNR})

LIEFERANT = ({LNR, FIRMA, ADR(PLZ, STADT, STRASSE), ANSPRECHP, GESCHLECHT }, {LNR})

BEDIENT = ({MNR, PNR}, {MNR})
BENÖTIGT = ({MNR, RNR, M_MENGE}, {MNR, RNR})
LIEFERT = ({RNR, LNR, L_MENGE}, {RNR})

5.3.4 Konsistenz im relationalen Modell

- intrarelationale Datenabhängigkeiten
- interrelationale Datenabhängigkeiten
- funktionale Datenabhängigkeiten
- Normalisierung

Definition:

Es sei $R = \{R_1, ..., R_k\}$ eine Menge von Relationenschemas, wobei $R_i = (X_i, \sum_i)$, $i=1(1)k$ und $X_i \neq X_j$ für $i \neq j$.

1. Eine relationale Datenbank d ist eine Menge von Relationen,

   $$d = \{r_1, ..., r_k\} \text{ mit } r_j \in Rel(X_i),\ i=1(1)k.$$

   Dat (R) Menge aller Datenbanken über R.

2. Eine Datenbank $d \in$ Dat (R) heißt punktweise konsistent, falls $r_i \in Sat(R_i)$ gilt für alle $r_i \in d$.

3. Sat (R) bezeichnet die Menge aller punktweise konsistenten Datenbanken über R.

4. R wird als Datenbankformat bezeichnet.

Beispiel einer Datenbank d mit Datenbankformat

R = {BUCH, LESER , ENTLIEHEN} und

buch $\in$ Sat (BUCH),

leser $\in$ Sat (LESER)

elhn $\in$ Sat (ENTLIEHEN)

buch:

INVNR	AUTOR	TITEL	VERLAG
1	Meier	Inf1	Springer
2	Meier	Inf2	Springer
3	Kron	DBS	IWT
4	Liskin	ORACLE	Hill

leser:

LNR	NAME	ADRESSE
500	Müller	Hamburg
550	Meier	Berlin
600	Schulz	Wedel

entliehen:

INVNR	LNR	RÜCKGABEDATUM
1	550	301110
2	550	301110
3	600	031210

Offen:
- Transformation der Komplexität bzw. des Typs einer Relationship-Deklaration
- Transformation einer hierarchischen Beziehung bzw. einer IS-A-Beziehung (Problem: implizite Existenzabhängigkeit)

Ausgedrückt im RDM unter Verwendung von **interrelationalen Datenabhängigkeiten.**

(Aussage darüber, wann eine Datenbank als „sinnvoll“ oder „gültig“ anzusehen ist)

Beispiel:
ELHN modelliert 1:n Beziehung zwischen LESER und BUCH.

d nur **gültig**, wenn

1. d punktweise konsistent ist,

2. kein Buch an mehr als einen Leser ausgeliehen ist, d.h. formal**:**

Ist $\mu \in$ ENTLIEHEN und μ[INVNR] = i ,
so existiert in ENTLIEHEN kein $v \neq \mu$ mit v[INVNR] = i .

Übertragung der Komplexitäts-Definition auf Relationen:

R=(X, $\sum$) sei Relationenschema, abgeleitet aus einer Relationship–Deklaration.

Sei **X = K_1 ... K_k Y**
mit **K_i** Primärschlüssel von R_i, i=1(1)k und **Y** Menge der R zusätzlich beschreibenden Attribute.

Sei **d** $\in$ **Dat**(R) und $r_1, r_2, \ldots, r_k, r \in d$.

In Analogie zum ERM wird definiert
comp(R, R_i) = (m, n) $\Leftrightarrow$

$$(\forall \mu \in r_i) : \left| \{ v \in r : v[K_i] = \mu[K_i] \} \right| \begin{cases} \geq m \\ \leq n \end{cases}$$

Definition: Abbildung σ^*

σ^* : **Dat**(R) $\rightarrow$ **{0, 1}**

mit {0, 1} als Menge der Boolschen Wahrheitswerte.

Falls σ^* (d) = 1 für d $\in$ Dat(R), so sagt man

d erfüllt Abhängigkeit σ^* oder **σ^* gilt in d.**

Definition: **Abbildung $\sum^*_R$**

$$\sum^*_R: \textbf{Dat}\,(R) \rightarrow \{0, 1\}$$

$$\text{mit } \sum^*_R(d) := \bigwedge_{\sigma^* \in \sum^*_R} \sigma^*(d)$$

$d \in Dat(R)$ erfüllt eine **Menge interrelationaler Abhängigkeiten,** falls d jede Abhängigkeit aus $\sum^*_R$ erfüllt, also $\sum^*_R(d) = 1$.

Definition: **Sat $(\sum^*_R)$: $\{d \in Dat(R) : \sum^*_R(d) = 1\}$**

Beispiel: Definition von Abbildungen σ^*_1 und σ^*_2 von Dat(R) in $\{0, 1\}$:

$$\sigma^*_1(d) := \begin{cases} 1 & \text{falls comp(ENTLIEHEN, BUCH)} \in \{(0, 1)\} \\ 0 & \text{sonst} \end{cases}$$

$$\sigma^*_2(d) := \begin{cases} 1 & \text{falls comp(ENTLIEHEN, LESER)} \in \{(0, k)\} \\ 0 & \text{sonst} \end{cases}$$

(k maximale Zahl von Büchern, die ein Leser zur gleichen Zeit ausgeliehen haben darf).

Mit $\sum^*_R = \{\sigma^*_1, \sigma^*_2\}$ als Menge **interrelationaler Abhängigkeiten** wird auch ausgedrückt, dass ein Buch an nicht mehr als einen Leser ausgeliehen sein darf.

Modellierung von **Existenzabhängigkeiten** in RDM:

Inklusionsabhängigkeiten

Beispiel: Sei buch, leser wie eben und

entliehen':

INVNR	LNR	RDAT
1	550	301210
6	550	061210

$\Rightarrow$ Buch entliehen, was in Relation b nicht vorkommt
$\Rightarrow$ Widerspruch zur Semantik dieser Beziehung im ERM.

d. h. zu jedem $\mu \in$ elhn muss ein $v \in$ b existieren mit **μ[INVNR] = v[INVNR],**

oder allgemein

zu jedem $\mu \in r$ muß ein $v \in r_i$ existieren mit $\mu[K_i] = v[K_i]$.

Definition: Sei R ein Datenbankformat,

$R_i, R_j \in R$, $R_i \neq R_j$, und
$R_i = (X_i, \Sigma_i)$
$R_j = (X_j, \Sigma_j)$
V sei Folge von n verschiedenen Attributen aus $\{X_i \cap X_j\}$, dabei sei $|\{X_i \cap X_j\}| \geq 1$.

Eine Inklusionsabhängigkeit (Inclusion Dependency -ID)

$$\mathbf{R_i\,[V] \subseteq R_j\,[V]}$$

bezeichnet folgende semantische Beziehung:
Sei $d \in Dat(R)$:

$$(R_i\,[V] \subseteq R_j\,[V])\,(d) := \begin{cases} 1 \text{ falls } \{\mu[V]:\mu \in r_i\} \subseteq \{\mu[V]:\mu \in r_j\} \\ 0 \text{ sonst} \end{cases}$$

IS-A-Beziehung führt auf ID,
aus Beziehung KB **IS-A** MITARBEITER wird
im RDM: KB[PNR] $\subseteq$ MITARBEITER[PNR]

Existenzabhängigkeit im Beispiel Bibliothek:
ENTLIEHEN [INVNR] $\subseteq$ BUCH [INVNR]
ENTLIEHEN [LNR] $\subseteq$ LESER [LNR]

Die Attribute INVNR und LNR heißen **Fremdschlüssel-Attribute** in **ENTLIEHEN.**

Allgemein wird definiert:

$K \subseteq X$ heißt **Fremdschlüssel** in $R = (X, \sum_x)$,

falls K in $R' \neq R$ **Primärschlüssel** ist und zu jedem Wert von K in der

R-Relation ein Primärschlüsselwert in der R´-Relation existiert.

$\Rightarrow$ **Fremdschlüssel sind allgemein als Inklusionsabhängigkeit $R[K] \subseteq R'[K]$ zu beschreiben.**

Für die Transformation 2-stelliger Beziehungen vom ERM in das RDM gilt:

Sei R eine **m:n-Beziehung** zwischen E_1 und E_2 und
seien R bzw. $R_i = (X_j, \{K_j\})$, i=1, 2 die zugehörigen Relationenschemas,
so wird die Gültigkeit der **ID**

$\mathbf{R[K_1] \subseteq R_1[K_1]}$ und
$\mathbf{R[K_2] \subseteq R_2[K_2]}$ gefordert.

fhwedel UNIVERSITY OF APPLIED SCIENCES

Definition:

Sei R eine Menge von **Relationenschemas** und $\sum^*_R$ eine Menge **interrelationaler** Abhängigkeiten.

1. Ein **Datenbankschema** ist ein Paar **D = (**R, $\sum^*_R$**).**

   Es dient der **zeitinvarianten konzeptuellen Beschreibung** der Menge

   Sat(D): = Sat(R, $\sum^*_R$)**: = Sat**(R **)** $\cap$ **Sat(**$\sum^*_R$)

   aller der Datenbanken über R, welche **alle intra- und interrelationalen Abhängigkeiten** erfüllen.

2. Eine Datenbank **d** $\in$ **Dat**(R**)** heißt **konsistent,** falls **d**$\in$ **Sat (D)** gilt.

   Jedes **d**$\in$ **Sat (D)** heißt **Beispiel für** oder **aktueller Zustand von D.**

fhwedel UNIVERSITY OF APPLIED SCIENCES

Konzeptionelle Datenmodellierung für ein betriebliches Informationssystem (10)

Noch nicht übertragen wurde die Information, dass **IS-A-Beziehungen** bestehen.
D.h. es bestehen **Existenzabhängigkeiten**, die als **interrelationale Datenabhängigkeiten** dargestellt werden können. Es muss also gelten:

MITARBEITER[PNR]	⊆	PERSON	[PNR]
ZEITAK[PNR]	⊆	PERSON	[PNR]
MANAGER[PNR]	⊆	MITARBEITER	[PNR]
KB [PNR]	⊆	MITARBEITER	[PNR]

Des weiteren gilt:

BEDIENT[MNR]	⊆	MASCHINE	[MNR]
BEDIENT[PNR]	⊆	MITARBEITER	[PNR]
BENÖTIGT [MNR]	⊆	MASCHINE	[MNR]
BENÖTIGT [RNR]	⊆	ROHSTOFF	[RNR]
LIEFERT [RNR]	⊆	ROHSTOFF	[RNR]
LIEFERT [LNR]	⊆	LIEFERANT	[LNR]

Man sieht:

MNR ist Fremdschlüssel in der Relation BEDIENT und BENÖTIGT,
PNR ist Fremdschlüssel in der Relation BEDIENT,
RNR ist Fremdschlüssel in der Relation BENÖTIGT und LIEFERT,
LNR ist Fremdschlüssel in der Relation LIEFERT.

Es sind folgende **Vereinfachungen** möglich:

Aufnahme von **PNR** in das **R-Schema** von MASCHINE:

→
MASCHINE = ({MNR, PNR, NAME, ANSCH_DATUM, NEUWERT, ZEITWERT}, {MNR})

Daraus folgt:

MASCHINE[PNR] $\subseteq$ **MITARBEITER [PNR]**

PNR ist **Fremdschlüssel** in der Relation MASCHINE.
Alle Informationen der Relation **BEDIENT** findet man jetzt in der Relation MASCHINE.

Eine solche Vereinfachung ist bezüglich der Relation **BENÖTIGT** nicht möglich, da dadurch eine **m:n-Beziehung** ausgedrückt wird und für jede **Kombination** der **Fremdschlüsselattribute** noch ein Wert des Attributes M_MENGE anzugeben ist.

Durch Aufnahme der Attribute LNR und L_MENGE in die Relation ROHSTOFF kann auf die Relation **LIEFERT** verzichtet werden:

ROHSTOFF = ({RNR, LNR, R_NAME, MENGE, PREIS, L_MENGE }, {RNR})

ROHSTOFF [LNR] $\subseteq$ **LIEFERANT[LNR].**

LNR ist **Fremdschlüssel** in der Relation ROHSTOFF.
Alle Informationen der Relation **LIEFERT** findet man jetzt in der Relation ROHSTOFF.

Es liegen die folgenden **R-Schemas** und **Abhängigkeiten** vor:

PERSON	=	({PNR, NAME, ADR(PLZ, STADT, STRASSE), TEL, E-MAIL}, {PNR})
MITARBEITER	=	({PNR, GEH_STUFE, BETRAG, ABT_NR, ABT_NAME, KRANKENKASSE, {P_BETRAG}, {KINDER(K_NAME, K_VORNAME, K_GEB)}}, {PNR})
ZEITAK	=	({PNR, L_FIRMA, H_LOHN, {VON_BIS}, H_SUMME}, {PNR})
MANAGER	=	({PNR, A_GRAD, STUD_RI}, {PNR})
KB	=	({PNR, KD_NAME}, {PNR})
MASCHINE	=	({MNR, PNR, NAME, ANSCH_DATUM, NEUWERT, ZEITWERT}, {MNR})
ROHSTOFF	=	({RNR, LNR, R_NAME, MENGE, PREIS, L_MENGE}, {RNR})
LIEFERANT	=	({LNR, FIRMA, ADR(PLZ, STADT, STRASSE), ANSPRECHP, GESCHLECHT}, {LNR})
BENÖTIGT	=	({MNR, RNR, M_MENGE }, {MNR, RNR})

MITARBEITER[PNR]	$\subseteq$	PERSON[PNR]
ZEITAK[PNR]	$\subseteq$	PERSON[PNR]
MANAGER[PNR]	$\subseteq$	MITARBEITER[PNR]
KB [PNR]	$\subseteq$	MITARBEITER[PNR]
MASCHINE[PNR]	$\subseteq$	MITARBEITER[PNR]
ROHSTOFF [LNR]	$\subseteq$	LIEFERANT[LNR]
BENÖTIGT [MNR]	$\subseteq$	MASCHINE[MNR]
BENÖTIGT [RNR]	$\subseteq$	ROHSTOFF[RNR]

fhwedel UNIVERSITY OF APPLIED SCIENCES

5.3.5 Funktionale Abhängigkeiten

5.3.5.1 Definitionen

Definition:

Sei V eine Attributmenge, $X, Y \subseteq V$.

Eine funktionale Abhängigkeit (Functional Dependency-FD)

$X \rightarrow Y$ bezeichnet folgende semantische Bedingung:

Sei $r \in Rel(V)$:

$$(X \rightarrow Y)(r): \begin{cases} 1 \text{ falls} & (\forall \mu, \nu \in r)(\mu[X] = \nu[X] \\ & \rightarrow \mu[Y]=\nu[Y] \\ 0 \text{ sonst} & \end{cases}$$

Schlüsselabhängigkeit ist eine spezielle FD.

Beispiel: **V = ABCDE und r** $\in$ Rel(V) und

r:

A	B	C	D	E
1	1	1	1	1
1	0	1	1	1
2	2	0	0	1
2	3	2	0	1

Menge F von FD's: $F = \{AB \rightarrow C, A \rightarrow D, D \rightarrow E\}$

Test, ob eine Relation $r \in Rel(V)$ eine FD der Form $X \rightarrow Y$ erfüllt:

- **Sortieren von r nach Werten des Attributes X,**
- **Prüfen aller Tupel mit gleichen X-Werten auf gleiche Y-Werte.**

FD's der Form $X \rightarrow Y$ erhalten „Namen“ $f: X \rightarrow Y$ oder einfach f.

L_f bzw. R_f bezeichnet die linke bzw. rechte Seite einer FD f, **attr(f) ist $L_f \cup R_f$.**

R = (V,F) mit FD-Menge F, wir betrachten die $r \in Rel(V)$, die F erfüllen.

Anstelle von Sat(R) wird auch Sat(F) geschrieben.

Bezüglich Update-Operationen stellen FDs spezielle Konsistenzbedingungen dar.

Diese müssen zur Laufzeit geprüft werden
$\Rightarrow$ **effiziente Prüfmechanismen nötig.**

Definition:
Seien F und G zwei FD-Mengen über einer Attributmenge V.

1. F und G heißen **äquivalent**, **$F \approx G$**, falls **Sat(F) = Sat(G)** gilt.
2. F heißt **redundant**, falls eine FD-Menge **$F' \subset F$** existiert mit **$F' \approx F$.**

Auf einen Begriff reduziert, ergibt sich die

Definition:

Sei F eine FD-Menge über V, f eine FD mit attr(f) $\subseteq$ V.
F impliziert f, kurz $F \longmapsto f$, falls Sat(F) $\subseteq$ Sat({f}) gilt.

D.h. jede Relation über V, welche F erfüllt, erfüllt auch f .

Satz:
Seien F und G wie eben.
Dann gilt

1. $F \approx G \quad \leftrightarrow (\forall f \in F)\ G \longmapsto f \wedge (\forall g \in G)\ F \longmapsto g$

2. **F ist redundant** $\leftrightarrow (\exists f \in F)\quad F-\{f\} \longmapsto f.$

fhwedel UNIVERSITY OF APPLIED SCIENCES

5.3.5.2 Das Membership-Problem

Ziel: Implikation „testen“
Dazu notwendig: Rechnen mit FDs

Definition:

Sei F eine Menge mit attr(f) $\subseteq$ V, dann heißt

$$\mathbf{F+ := \{ f : attr(f) \subseteq \bigcup_{g \in F} attr(g) \wedge F \mapsto f\}}$$

die Hülle (closure) von F.

Es folgt F $\subseteq$ F+ für jede FD-Menge f.

„Implikation testen“ bedeutet:

Ist f Element der Hülle einer FD-Menge F?

Zu entscheiden, ob zu gegebenem F und f gilt f $\in$ F+, heißt

Membership-Problem funktionaler Abhängigkeiten.

5.3.5.3 Basen von FDs

Gesucht: redundanzfreie Spezifikation von F

Definition:

Seien F und G FD-Mengen mit $F \approx G$.
Dann heißt **F Überdeckung von G**
(und G Überdeckung von F).

Eine FD-Menge F ist **redundant**, wenn es eine Teilmenge $F' \subset F$ gibt, welche F überdeckt.

Definition:

Sei F eine FD-Menge, $f : X \to Y \in F$:

1. f heißt **l(inks)-minimal**, falls kein $X' \subset X$ existiert mit $F - \{f\} \cup \{X' \to Y\} \approx F$.
2. f heißt **r(echts)-minimal**, falls $|Y| = 1$ gilt.
3. F heißt **minimal**, falls jedes Element von F l- und r-minimal ist.
4. Eine Überdeckung G von F heißt **Basis** von F, falls G nicht redundant und minimal ist.

fhwedel UNIVERSITY OF APPLIED SCIENCES

Konzeptionelle Datenmodellierung für ein betriebliches Informationssystem (9)

Übertragung des ER-Modells in das relationale Datenmodell: Ableitung von R-Schemas:

PERSON = ({PNR, NAME , ADR(PLZ,STADT, STRASSE), TEL, E-MAIL}, {PNR})

MITARBEITER = ({PNR, GEH_STUFE, BETRAG, ABT_NR, ABT_NAME, KRANKEN-KASSE, {P_BETRAG}, {KINDER (K_NAME, K_VORNAME, _GEB)}}, {PNR})

ZEITAK = ({PNR, L_FIRMA, H_LOHN, {VON_BIS}, H_SUMME}, {PNR})

MANAGER = ({PNR, A_GRAD, STUD_RI}, {PNR})

KB = ({PNR, KD_NAME}, {PNR})

MASCHINE = ({MNR, NAME, ANSCH_DATUM, NEUWERT, ZEITWERT}, {MNR})

ROHSTOFF = ({RNR, R_NAME, MENGE, PREIS}, {RNR})

LIEFERANT = ({LNR, FIRMA, ADR(PLZ, STADT, STRASSE), ANSPRECHP, GESCHLECHT }, {LNR})

BEDIENT = ({MNR, PNR}, {MNR})
BENÖTIGT = ({MNR, RNR, M_MENGE}, {MNR, RNR})
LIEFERT = ({RNR, LNR, L_MENGE}, {RNR})

fhwedel UNIVERSITY OF APPLIED SCIENCES

Konzeptionelle Datenmodellierung für ein betriebliches Informationssystem (11)

Untersuchung auf funktionelle Abhängigkeiten:

Relation PERSON:

PNR → NAME , ADR(PLZ, STADT, STRASSE), TEL, E-MAIL

PLZ → STADT

Relation MITARBEITER:

PNR → GEH_STUFE, BETRAG, ABT_NR, ABT_NAME, KRANKENKASSE {P_BETRAG}, {KINDER(K_NAME, K_VORNAME, K_GEB)}

GEH_STUFE → BETRAG

ABT_NR → ABT_NAME

Relation ZEITAK:

PNR → L_FIRMA, H_LOHN, {VON_BIS}, H_SUMME

Relation MANAGER:

PNR → A_GRAD , STUD_RI

Relation KB:

PNR → KD_NAME

Relation MASCHINE:

MNR → PNR, NAME, ANSCH_DATUM, NEUWERT, ZEITWERT

Relation ROHSTOFF:

RNR → LNR, R_NAME , MENGE , PREIS, L_MENGE

Relation LIEFERANT:

LNR → FIRMA , ADR(PLZ, STADT, STRASSE), ANSPRECHP, GESCHLECHT

PLZ → STADT

FIRMA → ADR(PLZ, STADT, STRASSE)

Relation BENÖTIGT:

MNR,RNR → M_MENGE

5.3.6 Entwurfs-Theorie relationaler DB-Schemas

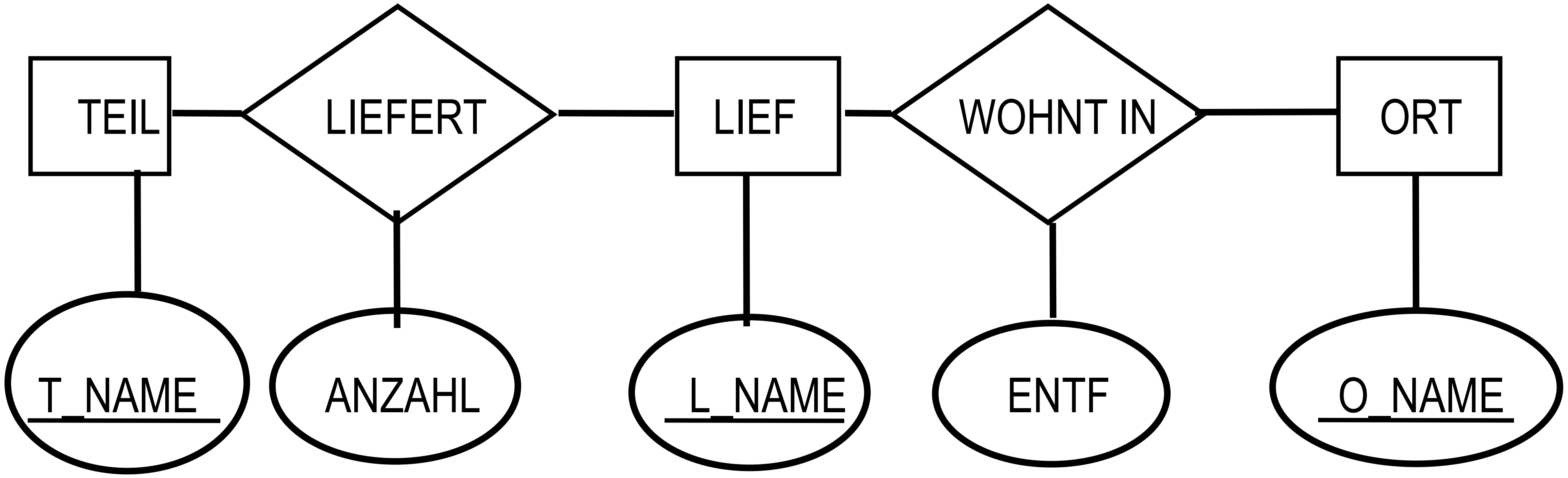
Ziel: logischer Entwurf eines relationalen Datenbankschemas

Beispiel:
Firma möchte **Daten speichern**

über ihre **Lieferanten** (LIEF)
- Namen (des Lieferanten) (L_NAME)
- Ort des Firmensitzes (O_NAME)
- die Entfernung von der Firma zu diesem Ort des Lieferanten (ENTF)

für jedes gelieferte **Teil** (TEIL)
- dessen Bezeichnung (T_NAME)
- dessen Anzahl (ANZAHL) .

Darstellung als ERD:



Transformation in das RDM:

LIEF = (L_NAME, { L_NAME → L_NAME})
TEIL = (T_NAME, { T_NAME → T_NAME})
ORT = (O_NAME, { O_NAME → O_NAME})

LIEFERT = (L_NAME, T_NAME, ANZAHL, { L_NAME, T_NAME → ANZAHL })
WOHNT IN = (L_NAME, O_NAME, ENTF, {L_NAME → O_NAME, ENTF; O_NAME → ENTF })

Ergebnis unbefriedigend-mit Redundanz behaftet.

Alternatives Vorgehen:

- Relationenschema mit allen Attributen einer Anwendung
- funktionale Abhängigkeiten

Das heißt:
Aufstellen eines Universalrelationen-Schema auf Basis eines Universum U (entspricht Diskursbereich).

Für jede funktionale Abhängigkeit gilt **attr(f)** $\subseteq$ **U.**
Ergebnis: Relationenschema der Form R = (U, F).

Ziel des logischen Entwurfs:
Schema zur Speicherung in einem Data Dictionary

5.3.6.1 Update-Anomalien

Universalschema für Beispiel:

R = **(U, F)** mit
U = L_NAME, T_NAME, ANZAHL, O_NAME, ENTF
F = **{** L_NAME, T_NAME $\rightarrow$ ANZAHL**;** L_NAME $\rightarrow$ O_NAME**,** O_NAME $\rightarrow$ ENTF **}**

r $\in$ **Rel(U)**

L_NAME	T_NAME	ANZAHL	O_NAME	ENTF
L1	T1	300	Berlin	300
L1	T2	200	Berlin	300
L1	T3	400	Berlin	300
L1	T4	200	Berlin	300
L1	T5	100	Berlin	300
L1	T6	100	Berlin	300
L2	T1	100	München	800
L2	T2	400	München	800
L3	T2	200	München	800
L4	T2	200	Bremen	100
L4	T4	300	Bremen	100
L4	T5	400	Bremen	100

Es gilt $\mathbf{r} \in \mathbf{Sat(\ R\ )}$, trotzdem **Anomalien**

1. **Einfüge-Anomalie**
   Neues Tupel erst eintragen, wenn alle 5 Informationen vorliegen.
2. **Lösch-Anomalie**
   Liefert Lieferant kein Teil mehr, geht auch Information über den Ort des Lieferanten verloren.
3. **Änderungs-Anomalie**
   Änderung eines Fakts bedeutet gegebenenfalls, in mehreren Tupeln zu ändern.

Basis: Normalformenlehre, d.h. aus einer Relation werden durch Normalisierung Relationen abgeleitet.

5.3.6.2 Normalformen

Erste Normalform:

Sicht auf die Wertebereiche der Attribute.

Es gilt:

- Eine Relation ist in **erster Normalform**, wenn die

  Wertebereiche aller Attribute nur **elementare Werte** enthalten.

Nicht erlaubt sind damit

- Werte, die selbst eine **Struktur** besitzen
- **Wiederholungsgruppen**.

Abhilfe durch **Normalisierung**.

Voraussetzung: Relationenschema in 1. NF

Definition:
Sei R = (U, F).
Ein Attribut A ∈ U heißt **prim**, falls es einen Schlüssel K für R gibt mit A ∈ K.

Beispiel:
R = (U, F) mit
U = (L_NAME, T_NAME, ANZAHL, O_NAME, ENTF)
F = {L_NAME, T_NAME → ANZAHL**;** L_NAME → O_NAME**;** O_NAME → ENTF}

L_NAME, T_NAME ist einziger Schlüssel
→ **L_NAME, T_NAME** sind prim,
ANZAHL, O_NAME, ENTF sind nichtprim.

Da **L_NAME, T_NAME** Schlüssel ist, folgt, dass aus F ableitbar ist:
L_NAME, T_NAME → ANZAHL**;**
L_NAME, T_NAME → O_NAME;
L_NAME, T_NAME → ENTF.

Diese FD's sind: -nicht trivial
r-minimal
nicht l-minimal
Es existieren Attribute, die **nichtprim** sind, aber voneinander abhängig.

Definition:

Sei R = (U, F).

$A \in U$ heißt **transitiv abhängig** von $X \subseteq U$, falls gilt

1. $X \to A \in F+$
2. $A \notin X$
3. $(\exists Y \subseteq U)\ [X \to Y \in F+ \wedge Y \to X \notin F+ \wedge Y \to A \in F+ \wedge A \notin Y]$

Definition:

Sei R = (U, F)

$A \in U$ heißt **direkt abhängig** von $X \subseteq U$, wenn

1. A nicht transitiv von X abhängig ist
2. $X \to A \in F+$

fhwedel UNIVERSITY OF APPLIED SCIENCES

Definition:

Sei R = (U, F) ein Relationenschema in 1. NF.

1. R ist in **2. NF**,
   falls für jedes Attribut $A \in U$ und A nichtprim und für jeden Schlüssel K für R gilt, dass die FD
   $K \rightarrow A$ I-minimal ist.

   (man sagt auch: A ist voll funktional abhängig von K und schreibt: $K \Rightarrow A$)

2. R ist in **3. NF**,
   falls für jedes Attribut $A \in U$ und A nicht prim und jeden Schlüssel K für R gilt, dass die FD
   $K \rightarrow A$ direkt ist.

Satz:

Sei R = (U, F) in 1. NF.

Dann gilt **R ist in 3. NF $\rightarrow$ R ist in 2. NF.**

Beispiel für funktionale Abhängigkeiten und Normalisierung:

Für den Aufbau eines weiteren **betrieblichen Informationssystems** wird eine Relation entworfen, die folgende Attribute besitzt:

PNR	Personalnummer jedes Mitarbeiters
NAME	Name des Mitarbeiters
VORNAME	Vorname des Mitarbeiters
VORWAHL	Vorwahlnummer des Arbeitsortes des Mitarbeiters
TEL#	Telefonnummer des Mitarbeiters
PLZ	PLZ des Arbeitsortes des Mitarbeiters
ORT	Arbeitsort des Mitarbeiters
STRASSE	Straße der Arbeitsstelle des Mitarbeiters
RAUM#	Raumnummer, in dem der Mitarbeiter seinen Arbeitsplatz hat
ABT#	Abteilungsnummer der Abteilung, in der der Mitarbeiter arbeitet
ABTNAME	Name der Abteilung, in der der Mitarbeiter arbeitet

Beobachtete Abhängigkeiten F:

PNR, NAME, VORNAME	→ VORWAHL, TEL#, PLZ, ORT, STRASSE
PNR	→ NAME, VORNAME
PNR	→ VORWAHL, TEL#, PLZ, ORT, STRASSE, RAUM#
PNR	**→ ABT#, ABTNAME**
ORT	→ VORWAHL
VORWAHL	→ ORT
NAME, VORNAME	→ RAUM#, ABT#, ABTNAME, PNR
TEL#	→ RAUM#
ABT#	→ ABTNAME
ABTNAME	→ ABT#
PLZ	→ ORT

Beobachtete Abhängigkeiten G:

PNR, NAME, VORNAME	→ VORWAHL, TEL#, PLZ, ORT, STRASSE
PNR	**→ NAME, VORNAME**
PNR	→ VORWAHL, TEL#, PLZ, ORT, STRASSE, RAUM#
ORT	→ VORWAHL
VORWAHL	→ ORT
NAME, VORNAME	→ RAUM#, **ABT#, ABTNAME,** PNR
TEL#	→ RAUM#
ABT#	→ ABTNAME
ABTNAME	→ ABT#
PLZ	→ ORT

F und G sind äquivalent.

F ist redundant.

PNR ➔ ABT#, ABTNAME kann abgeleitet werden.

G impliziert f : PNR ➔ ABT#, ABTNAME

PNR, NAME, VORNAME➔ VORWAHL, TEL#, PLZ, ORT , STRASSE

ist **nicht links-minimal**, denn es gilt bereits

PNR ➔ VORWAHL, TEL#, PLZ, ORT, STRASSE.

PNR ➔ VORWAHL, TEL#, PLZ, ORT, STRASSE ist **nicht rechts-minimal**

Die Abhängigkeiten

PNR ➔ VORWAHL
PNR ➔ TEL
PNR ➔ PLZ
PNR ➔ ORT
PNR ➔ STRASSE

sind **minimal**.

Entwicklung einer Basis:

PNR	➔	NAME
PNR	➔	VORNAME
PNR	➔	TEL#,
PNR	➔	PLZ
PNR	➔	STRASSE
PNR	➔	RAUM#
PNR	➔	ABT#
PNR	➔	ABTNAME
NAME, VORNAME	➔	PNR
ORT	➔	VORWAHL
VORWAHL	➔	ORT
TEL#	➔	RAUM#
ABT#	➔	ABTNAME
ABTNAME	➔	ABT#
PLZ	➔	ORT

Konstruktion von Relationen in 3.NF aus den **Basisabhängigkeiten**:

PNR als „Objekt“ + weitere Schlüssel + alle davon abhängigen Attribute, die nicht in weiteren funktionalen Abhängigkeiten auftauchen:

Also:

Personal = ({PNR, NAME, VORNAME, STRASSE},{PNR})

(Wir wissen: Alle von PNR funktional abhängigen Attribute hängen auch funktional von NAME, VORNAME ab.

Aus

PNR	➔	NAME
PNR	➔	VORNAME und
NAME, VORNAME	➔	PNR

folgt, dass bei Aufnahme von NAME, VORNAME in die Relation keine transitiven Abhängigkeiten entstehen können.)

ORTE als „Objekt" und VORWAHL als abhängiges Attribut:
ORTE = ({ORT, VORWAHL}, {ORT})

TELEFON als „Objekt" und RAUM# als abhängiges Attribut:
TELEFON = ({TEL#, RAUM#}, {TEL#})

PLZN als „Objekt" und ORT als abhängiges Attribut:
PLZN = ({PLZ, ORT }, {PLZ })

ABTEILUNG als „Objekt" und ABTNAME als abhängiges Attribut:
ABTEILUNG = ({ABT#, ABTNAME}, {ABT#})

Die folgenden Basisabhängigkeiten sind noch nicht realisiert:

PNR ➔ TEL#
PNR ➔ PLZ
PNR ➔ ABT#

Dabei stehen rechts nur Attribute, die **Primärschlüsselattribute** in anderen Relationen sind.

Diese gehen als **Fremdschlüssel** in die Relation **PERSONAL** ein:

PERSONAL = ({PNR, NAME, VORNAME, STRASSE, TEL#, PLZ, ABT#}, {PNR}).

Ergebnis:

PERSONAL	=	({PNR, NAME, VORNAME, STRASSE, TEL#, PLZ, ABT#}, {PNR})
ORTE	=	({ORT, VORWAHL}, {ORT})
TELEFON	=	({TEL#, RAUM#}, {TEL#})
PLZN	=	({PLZ, ORT}, {PLZ})
ABTEILUNG	=	({ABT#, ABTNAME}, {ABT#}).

Alle Relationen sind in 3.NF.

2. Weg:

Zerlegung der Relation Personal auf Grundlage des Studiums der funktionalen Abhängigkeiten.

Mit der Wahl von PNR als Primärschlüssel ist die Relation Personal automatisch in **zweiter Normalform.**

Sie ist **nicht in 3.NF**, weil transitive Abhängigkeiten existieren.

Die Attribute PNR, NAME, VORNAME sind **prim** (Schlüsselattribute).

Die Abhängigkeiten

TEL#	➔	RAUM#
ABT#	➔	ABTNAME
ABTNAME	➔	ABT#
ORT	➔	VORWAHL
VORWAHL	➔	ORT
PLZ	➔	ORT

im Bereich der **nichtprimen** Attribute führen zu **transitiven** Abhängigkeiten.

Diese Abhängigkeiten werden **durch Bildung eigener Relationen** aufgelöst und die jeweils rechts stehenden Attribute aus der großen Relation entfernt.

Gilt X ➔ Y und Y ➔ X , kann ausgewählt werden, welches Attribut den Primärschlüssel in der neuen Relation bildet und somit als Fremdschlüssel in der ursprünglichen Relation verbleibt.

Ergebnis:

PERSONAL = ({PNR, NAME, VORNAME, STRASSE, TEL#, PLZ, ABT#}, {PNR})

ORTE = ({ORT, VORWAHL}, {ORT})

TELEFON = ({TEL#, RAUM#}, {TEL#})

PLZN = ({PLZ, ORT }, {PLZ })

ABTEILUNG = ({ABT#, ABTNAME}, {ABT#})

Gesucht:

Test, um festzustellen, ob ein gegebenes **Relationenschema** einer Normalform genügt.

Aus dem folgenden Satz folgt, dass man zum Test auf 3. NF nicht überprüfen muss, ob **jedes nichtprime** Attribut von jedem Schlüssel **direkt** abhängt, sondern dass man sich auf einen **beliebig gewählten Schlüssel** beschränken kann.

Satz: Sei R = (U, F). Dann gilt

R ist in 3. NF ↔ jedes nichtprime Attribut direkt von irgend einem Schlüssel abhängig ist.

Testalgorithmus auf 3. NF für R = (U, F):

1. Man bestimme einen Schlüssel K für R.
2. Man bestimme alle nichtprimen Attribute von U.
3. Man prüfe, ob jedes nichtprime Attribut von K direkt abhängig ist.

5.3.6.3 Dekomposition und Synthese

Beispiel

R = (L_NAME, T_NAME, ANZAHL, O_NAME, ENTF,
{L_NAME, T_NAME $\rightarrow$ ANZAHL; L_NAME $\rightarrow$ O_NAME, O_NAME $\rightarrow$ ENTF})

R nicht in 2. NF.

Abhilfe:

R durch ein Datenbankschema ersetzen, welches die gleichen Anwendungen wie R modelliert, aber dafür besser geeignet ist. Ausgangspunkt ist die Basis der funktionalen Abhängigkeiten.

Wir betrachten das Datenbankformat

R' = {R1, R2, R2} mit

R1 = (L_NAME, T_NAME, ANZAHL,
{L_NAME, T_NAME $\rightarrow$ ANZAHL})

R2 = (L_NAME, O_NAME, {L_NAME $\rightarrow$ O_NAME})

R3 = (O_NAME, ENTF, {O_NAME $\rightarrow$ ENTF}),

$\rightarrow$

- R und R' „eng" verwandt
  (gleiche Attribute, gleiche Abhängigkeiten)
- jedes Schema von R' ist in 3.NF

Es gibt keine Anomalie und Redundanzen mehr.

fhwedel UNIVERSITY OF APPLIED SCIENCES

r1:

L_NAME	T_NAME	ANZAHL
L1	T1	300
L1	T2	200
L1	T3	400
L1	T4	200
L1	T5	100
L1	T6	100
L2	T1	100
L2	T2	400
L3	T2	200
L4	T2	200
L4	T4	300
L4	T5	400

r2:

L_NAME	O_NAME
L1	Berlin
L2	München
L3	München
L4	Bremen

r3:

O_NAME	ENTF
Berlin	300
München	800
Bremen	100

<u>Algorithmen:</u>

Algorithmus **Synthese:**

Input: R = (U,F)
Output: 3.NF Zerlegung von R
1. **Basis**: Erzeugung einer Basis für FDs
2. **Synthese**: Konstruktion von Relationen in 3. NF aus den Basisabhängigkeiten.

Algorithmus Dekomposition:

Input: R = (U,F) mit U Universalrelationenschema
Output: Zerlegung von R in 3.NF

Konzeptionelle Datenmodellierung für ein betriebliches Informationssystem (12)

Normalisierung:

1. Normalform:

Relation PERSON:

ADR ist **zusammengesetztes** Attribut, anstelle dieses Attributes nehmen wir die drei Attribute **PLZ**,**STADT**, **STRASSE** als **eigenständige** Attribute auf.

Relation **PERSON** ist damit in **erster NF**:

PERSON = ({PNR, NAME, PLZ, STADT, STRASSE, TEL, E-MAIL}, {PNR})

Alternativ:

Eigenständige Relation ADRESSE:

ADRESSE = ({ADR_ID, PLZ, STADT, STRASSE}, {ADR_ID})

Daraus ergibt sich als **Relation PERSON**:

PERSON = ({PNR, NAME ADR_ID, TEL, E-MAIL}, {PNR})

Relation MITARBEITER:

P_BETRAG ist **Wiederholungsgruppe**, wir schaffen **eigenständige Relation** Prämie mit den Attributen PNR und P_BETRAG.

Aus Diskussionen folgt, dass auch das **Datum** der Prämienzahlung von Interesse ist.

Daraus folgt **Relation**

PRÄMIE = ({PNR, P_DATUM, P_BETRAG},{PNR, P_DATUM })

und

MITARBEITER = ({PNR, GEH_STUFE, BETRAG, ABT_NR, ABT_NAME, KRANKENKASSE, {KINDER(K_NAME, K_VORNAME, K_GEB)}}, {PNR})

KINDER ist **zusammengesetztes Attribut** und **Wiederholungsgruppe**:

Wir schaffen **eigenständige Relation:**

KINDER = ({PNR, K_NAME, K_VORNAME, K_GEB}, {PNR, K_VORNAME})

und

MITARBEITER **=** ({PNR, GEH_STUFE, BETRAG, ABT_NR, ABT_NAME, KRANKENKASSE}, {PNR})

Relation ZEITAK :

VON_BIS ist Wiederholungsgruppe,
wir schaffen eine eigenständige Relation **EINSATZ**, wobei wir die Attribute trennen, also Attribute VON und BIS einführen.

EINSATZ = ({PNR, ENR, VON, BIS}, {PNR, ENR})

Zusätzlich wird das Attribut Einsatznummer ENR eingefügt. (Zählung erfolgt pro Zeitarbeitskraft)

ZEITAK = ({PNR, L_FIRMA, H_LOHN, H_SUMME}, {PNR})

Relation **LIEFERANT:**

LIEFERANT = ({LNR, FIRMA, ADR(PLZ, STADT, STRASSE), ANSPRECHP, GESCHLECHT}, {LNR})

ADR ist ein zusammengesetztes Attribut, wir nehmen die drei einzelnen Attribute auf:

LIEFERANT = ({LNR, FIRMA, PLZ, STADT, STRASSE, ANSPRECHP, GESCHLECHT}, {LNR})

Ergebnis: Alle Relationen sind jetzt in 1.NF.

fhwedel UNIVERSITY OF APPLIED SCIENCES

2. Normalform:

PERSON = ({PNR, NAME, PLZ, STADT, STRASSE, TEL, E-MAIL}, {PNR})
MITARBEITER = ({PNR, GEH_STUFE, BETRAG, ABT_NR, ABT_NAME, KRANKENKASSE}, {PNR})

PRÄMIE = ({PNR, P_DATUM, P_BETRAG}, {PNR, P_DATUM })
KINDER = ({PNR, K_NAME, K_VORNAME, K_GEB}, {PNR, K_VORNAME})
ZEITAK = ({PNR, L_FIRMA, H_LOHN, H_SUMME}, {PNR})
EINSATZ = ({PNR, ENR, VON, BIS}, {PNR, ENR})

MANAGER = ({PNR, A_GRAD, STUD_RI}, {PNR})
KB = ({PNR, KD_NAME}, {PNR})

MASCHINE = ({MNR, PNR, NAME, ANSCH_DATUM, NEUWERT, ZEITWERT}, {MNR})
ROHSTOFF = ({RNR, LNR, R_NAME, MENGE, PREIS, L_MENGE}, {RNR})
LIEFERANT = ({LNR, FIRMA, PLZ, STADT, STRASSE, ANSPRECHP, GESCHLECHT }, {LNR})

BENÖTIGT = ({MNR, RNR, M_MENGE}, {MNR, RNR})

Es sind nur die Relationen zu prüfen, deren Primärschlüssel aus mindestens 2 Attributen zusammengesetzt sind.

Es gilt:

PNR, P_DATUM $\Rightarrow$ P_BETRAG
(I-minimal, voll funktional abhängig)

PNR , K_VORNAME $\Rightarrow$ K_NAME, K_GEB

PNR, ENR $\Rightarrow$ VON, BIS

MNR,RNR $\Rightarrow$ M_MENGE

fhwedel UNIVERSITY OF APPLIED SCIENCES

3. Normalform :

Wir suchen **funktionale Abhängigkeiten** im Bereich der **Nichtschlüsselattribute**:

Dazu betrachten wir die funktionalen Abhängigkeiten.

Relation PERSON:

PERSON = ({PNR, NAME, PLZ, STADT, STRASSE,TEL, E-MAIL}, {PNR})
PLZ → STADT

Daraus eigene Relation **STADT** mit den Attributen PLZ, STADT:

STADT = ({PLZ, STADT}, { PLZ})
PERSON = ({PNR, NAME, PLZ, STRASSE, TEL, E-MAIL}, {PNR})

Relation MITARBEITER:

MITARBEITER = ({PNR, GEH_STUFE, BETRAG, ABT_NR, ABT_NAME, KRANKENKASSE}, {PNR})
GEH_STUFE → BETRAG
ABT_NR → ABT_NAME

Eigene Relationen:

GEHALT = ({GEH_STUFE, BETRAG}, {GEH_STUFE})
ABTEILUNG = ({ABT_NR, ABT_NAME}, {ABT_NR})
MITARBEITER = ({PNR, GEH_STUFE, ABT_NR, KRANKENKASSE}, {PNR})

Relation Prämie:

PRÄMIE = ({PNR, P_DATUM, P_BETRAG},{PNR, P_DATUM})

Automatisch 3.NF (nur ein Nichtschlüsselattribut)

Relation KINDER:
KINDER = ({PNR, K_NAME, K_VORNAME, K_GEB }, {PNR, K_VORNAME })

Relation ZEITAK:
ZEITAK = ({PNR, L_FIRMA, H_LOHN, H_SUMME}, {PNR})

Frage: Hängt H_LOHN von der PERSON oder der L_FIRMA ab?
Wenn von PERSON abhängig → 3.NF,
wenn von L_FIRMA abhängig →Normalisierung
Z1=({PNR, L_FIRMA, H_SUMME},{PNR}) und
Z2= ({L_FIRMA, H_LOHN},{L_FIRMA})

Relation Einsatz:
EINSATZ = ({PNR, ENR, VON, BIS}, {PNR, ENR})

Relation MANAGER:
MANAGER = ({PNR, A_GRAD, STUD_RI}, {PNR})

Relation KB:
KB = ({PNR, KD_NAME}, {PNR})

Relation MASCHINE:
MASCHINE = ({MNR, PNR, NAME, ANSCH_DATUM, NEUWERT, ZEITWERT}, {MNR})

Relation ROHSTOFF:
ROHSTOFF = ({RNR, LNR, R_NAME, MENGE, PREIS, L_MENGE}, {RNR})

Relation LIEFERANT:

LIEFERANT = ({LNR, FIRMA, PLZ, STADT, STRASSE, ANSPRECHP, GESCHLECHT}, {LNR})

PLZ→STADT

LIEFERANT = ({LNR, FIRMA, PLZ, STRASSE, ANSPRECHP, GESCHLECHT }, {LNR})

Relation BENÖTIGT:

BENÖTIGT = ({MNR, RNR, M_MENGE}, {MNR, RNR})

Inklusionsabhängigkeiten

MITARBEITER[PNR]	$\subseteq$	PERSON[PNR] I
ZEITAK[PNR]	$\subseteq$	PERSON[PNR]
MANAGER[PNR]	$\subseteq$	MITARBEITER[PNR]
KB [PNR]	$\subseteq$	MITARBEITER[PNR]
PRAEMIE [PNR]	$\subseteq$	MITARBEITER[PNR]
KINDER [PNR]	$\subseteq$	MITARBEITER[PNR]
EINSATZ [PNR]	$\subseteq$	ZEITAK[PNR]
MASCHINE[PNR]	$\subseteq$	MITARBEITER[PNR]
ROHSTOFF [LNR]	$\subseteq$	LIEFERANT[LNR]
BENÖTIGT [MNR]	$\subseteq$	MASCHINE [MNR]
BENÖTIGT [RNR]	$\subseteq$	ROHSTOFF [RNR]
PERSON[PLZ]	$\subseteq$	STADT[PLZ]
MITARBEITER[GEH_STUFE]	$\subseteq$	GEHALT[GEH_STUFE]
MITARBEITER[ABT_NR]	$\subseteq$	ABTEILUNG[ABT_NR]

Beispiel für Normalisierung:

Im Rahmen einer UNO-Erhebung werden erfasst:

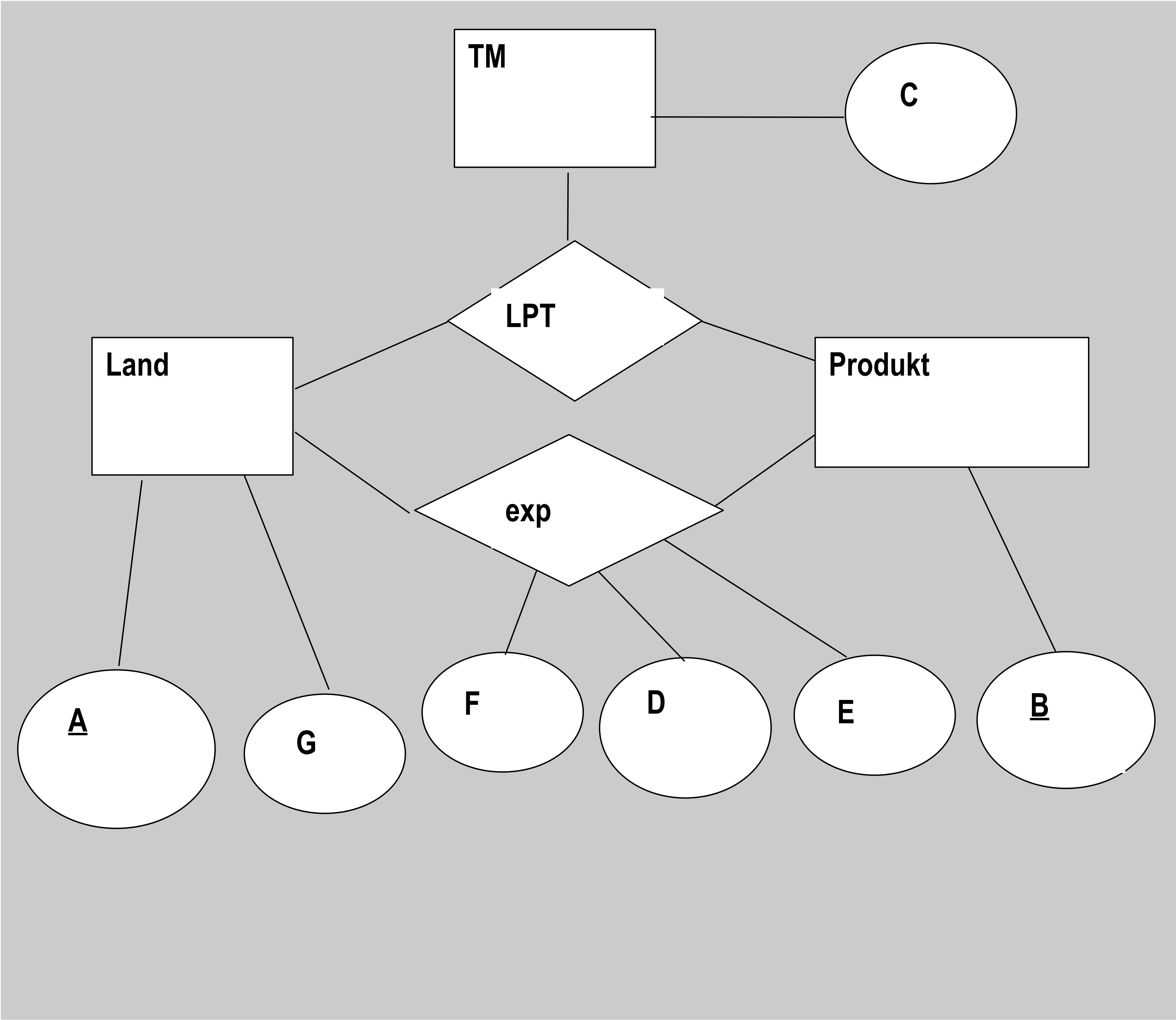
Von jedem Land der **Name** (A) und
die **Einwohnerzahl** in Millionen (G),
die **Produkte** (B), die exportiert werden und
das **Transportmittel** (C), das für den Transport eines jeden Produktes durch das Land eingesetzt wird,
den **Preis** (D), den das Land beim Export für jedes Produkt erzielt,
die verkaufte **Menge** (E) eines jeden Produktes,
die gezahlte **Versicherungssumme** (F) für den Transport.

Die erfassten Werte sind in der folgenden Relation r aufgeführt.

Wie lauten die funktionalen Abhängigkeiten?

1. Leiten Sie dazu die funktionalen Abhängigkeiten allein aus der Analyse der Werte ab.
2. Überprüfen Sie, ob die ermittelten funktionalen Abhängigkeiten auch die Realität richtig widerspiegeln.

Leiten Sie daraus Relationen in 3.NF ab.


TM
C
LPT
Land
Produkt
exp
A
G
F
D
E
B

Gegeben sei die folgende Relation r:

A	B	C	D	E	F	G
a_1	b_1	c_1	d_1	e_1	f_1	g_5
a_1	b_2	c_1	d_1	e_2	f_1	g_5
a_1	b_3	c_1	d_1	e_1	f_1	g_5
a_1	b_4	c_1	d_4	e_1	f_4	g_5
a_1	b_5	c_2	d_2	e_2	f_4	g_5
a_2	b_1	c_1	d_2	e_2	f_4	g_4
a_2	b_2	c_2	d_2	e_3	f_4	g_4
a_3	b_1	c_1	d_3	e_4	f_6	g_3
a_3	b_1	c_6	d_3	e_4	f_6	g_3
a_3	b_1	c_7	d_3	e_4	f_6	g_3
a_3	b_2	c_3	d_3	e_5	f_6	g_3
a_4	b_1	c_2	d_3	e_2	f_6	g_3

A	B	C	D	E	F	G
a_1	b_1	c_1	d_1	e_1	f_1	g_5
a_1	b_2	c_1	d_1	e_2	f_1	g_5
a_1	b_3	c_1	d_1	e_1	f_1	g_5
a_1	b_4	c_1	d_4	e_1	f_4	g_5
a_1	b_5	c_2	d_2	e_2	f_4	g_5
a_2	b_1	c_1	d_2	e_2	f_4	g_4
a_2	b_2	c_2	d_2	e_3	f_4	g_4
a_3	b_1	c_1	d_3	e_4	f_6	g_3
a_3	b_1	c_6	d_3	e_4	f_6	g_3
a_3	b_1	c_7	d_3	e_4	f_6	g_3
a_3	b_2	c_3	d_3	e_5	f_6	g_3
a_4	b_1	c_2	d_3	e_2	f_6	g_3

Feststellen von funktionalen Abhängigkeiten:
A → G
D → F
A, B → D,E, F
A,B,C → D,E, F, G

⇒ **Primärschlüssel: K = (A,B,C) ,**
⇒ **A,B,C prim**

Man sieht sofort: nur in 1. NF

Daraus folgen Relationen in 2. NF:

<u>A</u>	<u>B</u>	D	E	F
a_1	b_1	d_1	e_1	f_1
a_1	b_2	d_1	e_2	f_1
a_1	b_3	d_1	e_1	f_1
a_1	b_4	d_4	e_1	f_4
a_1	b_5	d_2	e_2	f_4
a_2	b_1	d_2	e_2	f_4
a_2	b_2	d_2	e_3	f_4
a_3	b_1	d_3	e_4	f_6
a_3	b_2	d_3	e_5	f_6
a_4	b_1	d_3	e_2	f_6

<u>A</u>	G
a_1	g_5
a_2	g_4
a_3	g_3
a_4	g_3

fhwedel UNIVERSITY OF APPLIED SCIENCES

Daraus folgen Relationen in 3. NF:

A	B	D	E
a_1	b_1	d_1	e_1
a_1	b_2	d_1	e_2
a_1	b_3	d_1	e_1
a_1	b_4	d_4	e_1
a_1	b_5	d_2	e_2
a_2	b_1	d_2	e_2
a_2	b_2	d_2	e_3
a_3	b_1	d_3	e_4
a_3	b_2	d_3	e_5
a_4	b_1	d_3	e_2

A	G
a_1	g_5
a_2	g_4
a_3	g_3
a_4	g_3

D	F
d_2	f_1
d_3	f_4
d_3	f_6
d_3	f_4