

## 2. Logischer DB-Entwurf

- **Vorgehensweisen beim Entwurf eines relationalen Schemas**
  - Normalisierung
  - Synthese
- **Grundlagen, Definitionen und Begriffe**
  - Funktionale Abhängigkeiten, Schlüssel
  - Axiomensystem nach Armstrong
  - Bestimmung funktionaler Abhängigkeiten
  - Zerlegung von Relationen
  - Abhängigkeitsbewahrung
- **Normalformenlehre**
  - Erste Normalform (1NF), 2NF, 3NF
  - BCNF und 4NF
  - 5NF und weitere (nicht behandelt)
- **Entwurfstheorie für relationale Datenbanken**
  - Membership-Problem
  - Minimale Überdeckungen
- **Synthese von Relationen**
  - Synthese-Algorithmus von Beeri/Bernstein
  - Beispiele
  - Berücksichtigung von Anwendungsaspekten

# Entwurf eines relationalen DB-Schemas

- **Ziel:**

Theoretische Grundlage für den Entwurf eines „guten“ relationalen DB-Schemas (→ Entwurfstheorie, Normalisierungslehre)

- **Güte:**

- leichte Handhabbarkeit, Verständlichkeit, Natürlichkeit, Übersichtlichkeit, ...
- Entwurfstheorie versucht „Güte“ zu präzisieren/formalisieren

- **Beispiele**

KunterBunt (A1, A2, A3, ..., A300)

ABTMGR (ANR, ANAME, BUDGET, MNR, PNAME, TITEL, SEIT\_JAHR)

- **Was macht einen schlechten DB-Schema-Entwurf aus?**

- implizite Darstellung von Informationen
- Redundanzen, potentielle Inkonsistenz (Änderungsanomalien)
- Einfügeanomalien, Löschanomalien
- ...

→ oft hervorgerufen durch „Vermischung“ von Entities, Zerlegung und wiederholte Speicherung von Entities, ...

- **Normalisierung von Relationen**

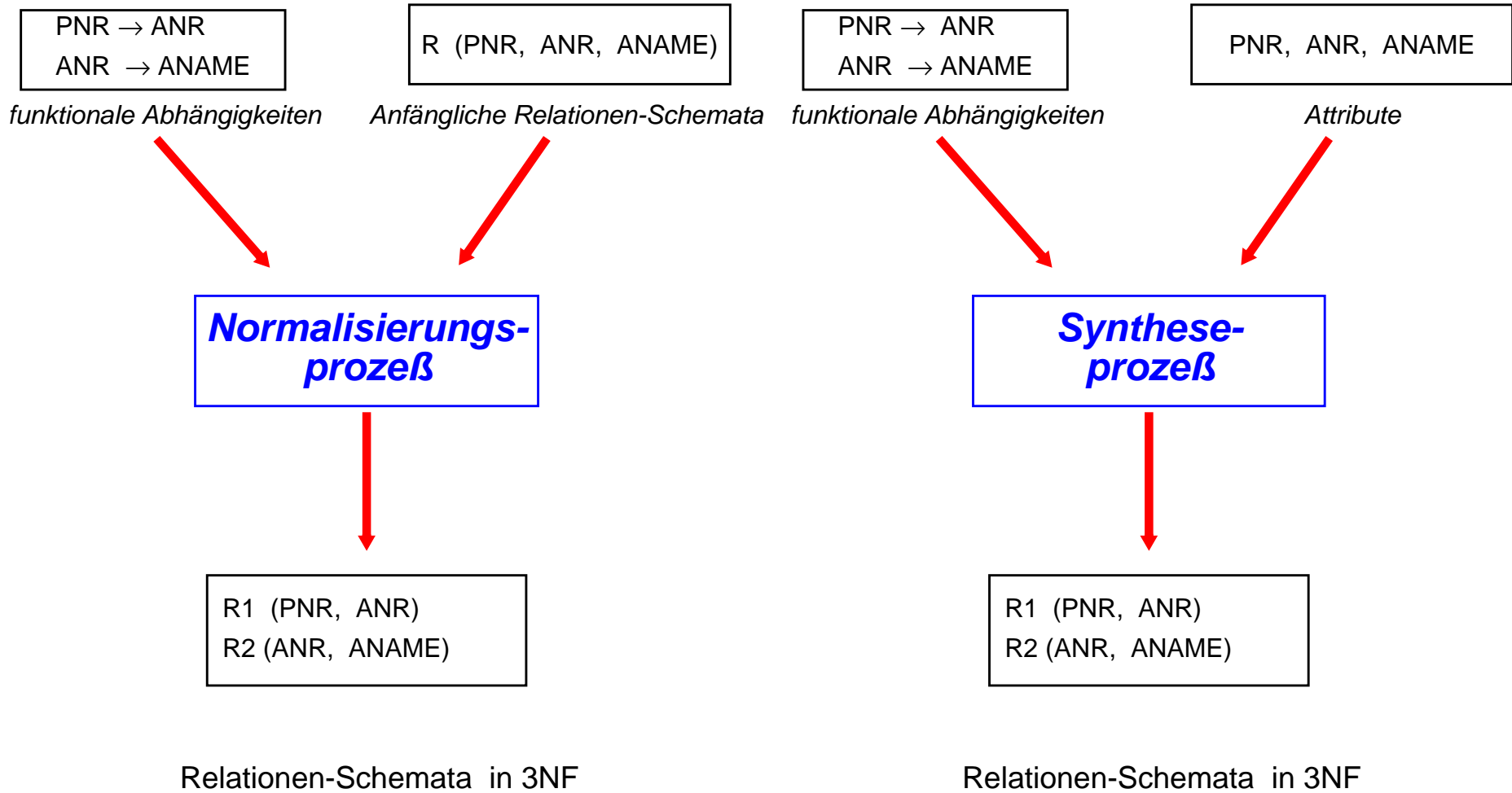
hilft einen gegebenen Entwurf zu verbessern

- **Synthese von Relationen**

zielt auf die Konstruktion eines „optimalen“ DB-Schemas ab

## Normalisierung von Relationen

## Synthese von Relationen



# Funktionale Abhängigkeit

- Konventionen:**

$R, S$	Relationenschemata (Relationenname, Attribute)
$R, S$	Relationen der Relationenschemata $R, S$
$A, B, C, \dots$	einfache Attribute
$A = \{A_1, \dots, A_n\}$	Attributmenge des Relationenschemas
$W, X, Y, Z, \dots$	Mengen von Attributen
$a, b, c$	Werte einfacher Attribute
$x, y, z$	Werte von $X, Y, Z$
$XY \equiv X \cup Y$	Mengen brauchen nicht disjunkt zu sein

- Definition: Funktionale Abhängigkeit (FA)**

(engl. functional dependency)

Die FA  $X \rightarrow Y$  gilt ( $X$  bestimmt  $Y$  funktional), wenn für alle  $R$  von  $\mathcal{R}$  gilt: Zwei Tupel, deren Komponenten in  $X$  übereinstimmen, stimmen auch in  $Y$  überein.

$$\forall t \in R \quad \forall u \in R \quad (t.X = u.X) \Rightarrow (t.Y = u.Y)$$

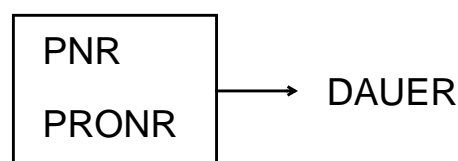
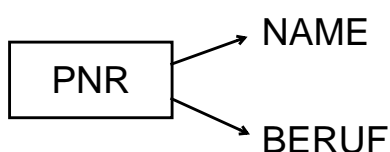
*alternativ:*

Die Relation  $R$  erfüllt die FA  $X \rightarrow Y$ , wenn für jeden  $X$ -Wert  $x$   $\pi_Y(\sigma_{X=x}(R))$  höchstens ein Tupel hat.

- Notation:**

$\{PNR\} \rightarrow \{NAME, BERUF\}$ : verkürzt  $PNR \rightarrow NAME, BERUF$

$\{PNR, PRONR\} \rightarrow \{DAUER\}$ : verkürzt  $PNR, PRONR \rightarrow DAUER$



## Funktionale Abhängigkeit (2)

- **Beispiel**

Gegeben sei die Relation  $R$  mit dem Schema  $R = \{A, B, C, D\}$  und der FA  $A \rightarrow B$ .

R	A	B	C	D
	$a_1$	$b_1$	$c_1$	$d_1$
	$a_1$	$b_1$	$c_1$	$d_2$
	$a_2$	$b_2$	$c_3$	$d_2$
	$a_3$	$b_2$	$c_4$	$d_3$
	$a_4$	$b_2$	$c_4$	$d_3$

Welche weiteren FA's erfüllt die gezeigte Relation  $R$ ?

- **Triviale funktionale Abhängigkeit**

Funktionale Abhängigkeiten, die von jeder Relationenausprägung automatisch immer erfüllt sind, nennt man triviale FA's.

Nur FA's der Art  $X \rightarrow Y$  mit  $Y \subseteq X$  sind trivial.

Es gilt also  $R \rightarrow R$

- **Achtung:**

- FA's lassen sich nicht durch Analyse einer Relation  $R$  gewinnen. Sie sind vom Entwerfer festzulegen.
- FA's beschreiben semantische Integritätsbedingungen bezüglich der Attribute eines Relationenschemas, die jederzeit erfüllt sein müssen

# Schlüssel

- **Superschlüssel**

- Im Relationenschema  $R$  ist  $X \subseteq R$  ein Superschlüssel, falls gilt:

$$X \rightarrow R$$

- Falls  $X$  Schlüsselkandidat von  $R$ , dann gilt für alle  $Y$  aus  $R$ :

$$X \rightarrow Y$$

➔ Wir benötigen das Konzept der vollen funktionalen Abhängigkeit, um Schlüssel (-kandidaten) von Superschlüsseln abzugrenzen.

- **Volle funktionale Abhängigkeit**

$Y$  ist voll funktional abhängig ( $\Rightarrow$ ) von  $X$ , wenn gilt

1.  $X \rightarrow Y$ ,
2.  $X$  ist „minimal“, d. h.  $\forall A_i \in X : X - \{A_i\} \not\rightarrow Y$

➔  $Y$  ist funktional abhängig von  $X$ , aber nicht funktional abhängig von einer echten Teilmenge von  $X$ . Falls  $X \Rightarrow R$  gilt, bezeichnet man  $X$  als Schlüsselkandidat von  $R$

- **Beispiel**

Eine Stadt werde beschrieben durch Name, BLand (Bundesland), EW (Einwohnerzahl) und VW (Vorwahl)

Stadt	Name	BLand	EW	VW
	K'lautern	Rlp	100 000	0631
	Mainz	Rlp	250 000	06131
	Frankfurt	Bdg	90 000	0335
	Frankfurt	Hes	700 000	069
	...			

- Superschlüssel

- Schlüsselkandidaten

# Bestimmung funktionaler Abhängigkeiten

- **Informationsbedarfsanalyse liefert**

- Menge aller Attribute (bei existierenden Relationen bereits vorgegeben)
- Menge  $F$  der funktionalen Abhängigkeiten zwischen Attributen

↳ **Achtung:**  $F$  kann redundante FA's enthalten! Falls jedoch eine nicht-redundante FA übersehen wurde, ist diese Information „nicht-existent“

- **Beispiel**

- Attribute: PNR, SVNR, BERUF, ANR, AORT
- Menge  $F$  der FA's:

1. SVNR  $\rightarrow$  BERUF
2. PNR  $\rightarrow$  SVNR, ANR
3. SVNR, BERUF  $\rightarrow$  PNR
- 4.

Gilt SVNR  $\rightarrow$  ANR?      Gilt ANR  $\rightarrow$  AORT?

- **Definition: *Logische Implikation***

Sei  $F$  eine Menge von FA's für  $R$  und sei  $X \rightarrow Y$  eine FA.

Dann impliziert  $F \models X \rightarrow Y$  logisch ( $F \models X \rightarrow Y$ ), wenn jedes  $R$  aus  $R$ , das die FA's in  $F$  erfüllt, auch  $X \rightarrow Y$  erfüllt.

## Bestimmung funktionaler Abhängigkeiten (2)

- **Axiome für die Ableitung funktionaler Abhängigkeiten**

➔ Inferenzregeln zum Ableiten von FA's aus Menge gegebener FA's

- **Axiomensystem nach Armstrong**

A1: (Reflexivität):

Wenn  $Y \subseteq X \subseteq A$ , dann  $X \rightarrow Y$

A2: (Verstärkung):

$X \rightarrow Y \models XW \rightarrow YZ$  ( $Z \subseteq W \subseteq A$ )

A3: (Transitivität):

$X \rightarrow Y, Y \rightarrow Z \models X \rightarrow Z$

- **Veranschaulichung**

$X \rightarrow Y$ , d. h.  $\forall t \in R \forall u \in R (t.X = u.X) \Rightarrow (t.Y = u.Y)$

PERS ( ...	PNR,	BERUF,	ANR,	AORT )
...	123	Prog	A1	KL
...	456	Op	A2	SB
...	789	Op	A1	KL
...	333	Mgr	A3	KL
...	123	Prog	A1	KL



## Bestimmung funktionaler Abhängigkeiten (3)

- **Weitere Regeln:**

Es ist für den Herleitungsprozeß hilfreich, noch 3 weitere Regeln einzusetzen

R4: (Vereinigung):

$$X \rightarrow Y, X \rightarrow Z \models X \rightarrow YZ$$

R5: (Zerlegung):

$$X \rightarrow YZ \models X \rightarrow Y$$

R6: (Pseudotransitivität):

$$X \rightarrow Y, YW \rightarrow Z \models XW \rightarrow Z$$

- **Einsatz von R4 und R5**

## Bestimmung funktionaler Abhängigkeiten (4)

- **Definition:** *Hülle von F*

$F^+$  ist die Menge der FA's, die logisch durch F impliziert werden:

$$F^+ = \{X \rightarrow Y \mid F \models X \rightarrow Y\}$$

➔  $F^+$  lässt sich mittels der drei Armstrong-Axiome ableiten.

- **Bemerkung:**  $F^+ = (F^+)^+$

- **Satz:**

Sei F die Menge von FA's für Relationenschema  $R$  und  $X \rightarrow Y$  eine weitere FA für  $R$ . Dann gilt:

$X \rightarrow Y$  ist genau dann in  $F^+$  enthalten, wenn für jede Ausprägung  $R$  von  $R$ , die alle FA's aus F erfüllt, auch  $X \rightarrow Y$  erfüllt.

- **Beispiel:** Tabelle PERS

F: PNR  $\rightarrow$  BERUF, PNR  $\rightarrow$  ANR, ANR  $\rightarrow$  AORT

- **Axiome A1 – A3 sind vollständig und korrekt**

- vollständig: Aus F lassen sich alle FA's in  $F^+$  ableiten
- korrekt (*sound*): Es wird aus F keine FA abgeleitet, die nicht in  $F^+$  ist

## Bestimmung funktionaler Abhängigkeiten (5)

- Beispiel zur Miniwelt „Universität“

- Attribute: PNR, PNAME, FACH, NOTE, PDAT  
MATNR, NAME, GEB, ADR, FBNR, FBNAME, DEKAN

- Menge F der FA's:

1. PNR  $\rightarrow$  PNAME, FACH
2. MATNR  $\rightarrow$  NAME, GEB, ADR, FBNR
3. NAME, GEB, ADR  $\rightarrow$  MATNR
4. PNR, MATNR, FBNR  $\rightarrow$  NOTE, PDAT
5. FBNR  $\rightarrow$  FBNAME
6. DEKAN  $\rightarrow$  FBNR, FBNAME
7. FBNAME  $\rightarrow$  DEKAN, FBNR

Ist MATNR  $\rightarrow$  DEKAN ableitbar?

Ist PNR, MATNR, FBNR Schlüsselkandidat?

## Bestimmung funktionaler Abhängigkeiten (6)

- **Definition: Hülle einer Attributmenge**

$X^+$  ist **Hülle einer Attributmenge**  $X$  bzgl.  $F$ .

Sie umfaßt die Menge aller Attribute  $A_i$ , für die  $X \rightarrow A_i$  in  $F^+$  ist.

- **Satz:** Test, ob  $FA$  in  $F^+$

$X \rightarrow Y$  ist in  $F^+$  genau dann, wenn  $Y \subseteq X^+$

- **Bemerkung:** Satz liefert **einfache Möglichkeit** zu entscheiden, ob eine vorgegebene  $FA$  in  $F^+$  ist oder nicht

- **Algorithmus CLOSURE**

Einfacher Algorithmus zur Bestimmung von  $X^+$ ; ( $F^+$  schwierig)

Eingabe: Menge von Attributen  $X$  und Menge  $F$  von  $FA$ 's

Ausgabe:  $X^+$  bzgl.  $F$

CLOSURE( $X, F$ )

begin

    OLDDEP :=  $\emptyset$ ; NEWDEP :=  $X$ ;

while NEWDEP  $\neq$  OLDDEP do

begin

            OLDDEP := NEWDEP;

for jede  $FA$   $Y \rightarrow Z$  in  $F$  do

if  $Y \subseteq$  NEWDEP then

                    NEWDEP := NEWDEP  $\cup$   $Z$

end

return (NEWDEP)

end

## Bestimmung funktionaler Abhängigkeiten (7)

- **Zwei Aufgaben:**

1. Berechnung von  $F^+$  aus  $F$  (wird praktisch nicht durchgeführt)
2. Gegeben  $F$ ,  $X \rightarrow Y$ :

Ist  $X \rightarrow Y$  in  $F^+$  ?

Ist  $X \rightarrow Y$  in  $(F - \{X \rightarrow Y\})^+$  ?

- **Algorithmus MEMBER: Testen auf Mitgliedschaft**

Eingabe:  $X \rightarrow Y$ ,  $F$

Ausgabe: TRUE, wenn  $F \models X \rightarrow Y$ , sonst FALSE

MEMBER ( $F$ ,  $X \rightarrow Y$ )

begin

if  $Y \subseteq \text{CLOSURE}(X, F)$  then

return (TRUE)

else return (FALSE)

end

# Zerlegung von Relationen

- „Schlechtes“ Relationsschema

PRÜFUNGEN (	PNR,	PNAME,	FACH,	MATNR,	NAME,	FB,	NOTE )
	1	Härder	DBS	1234	Müller	Inf	1
	1	Härder	DBS	5678	Maier	Mathe	2
	1	Härder	DBS	9000	Schmitt	Inf	3
	2	Schock	FA	5678	Maier	Mathe	4
	2	Schock	FA	0007	Coy	Mathe	2

- Drei Arten von Anomalien

- Änderungsanomalien

- erhöhter Speicherplatzbedarf wegen redundant gespeicherter Information
- gleichzeitige Aktualisierung aller redundanten Einträge erforderlich!
- Leistungseinbußen, da mehrere redundante Einträge geändert werden müssen

- Einfüge- und Löschanomalien

- Vermischung von Informationen zweier Entity-Typen führt auf Probleme, wenn Information eingetragen/gelöscht werden soll, die nur zu einem der Entity-Typen gehört
- Erzeugen vieler NULL-Werte oder Verlust von Information

## Zerlegung von Relationen (2)

- **Anomalien** sind darauf zurückzuführen, daß „**nicht zusammenpassende**“ Informationen vermischt werden
- **Grundlegende Korrektheitskriterien** für eine Zerlegung oder Normalisierung von Relationenschemata

1. **Verlustlosigkeit**

Die in der ursprünglichen Ausprägung  $R$  des Schemas  $R$  enthaltenen Informationen müssen aus den Ausprägungen  $R_1, \dots, R_n$  der neuen Relationenschemata  $R_1, \dots, R_n$  rekonstruierbar sein.

2. **Abhängigkeitsbewahrung**

Die für  $R$  geltenden funktionalen Abhängigkeiten müssen auf die Schemata  $R_1, \dots, R_n$  übertragbar sein.

- **Gültige Zerlegung**

$$R = R_1 \cup R_2,$$

d.h., alle Attribute aus  $R$  bleiben erhalten

- **Verlustlose Zerlegung**

$$R_1 := \Pi_{R_1}(R)$$

$$R_2 := \Pi_{R_2}(R)$$

wenn für jede mögliche (gültige) Ausprägung  $R$  von  $R$  gilt

$$R = R_1 \bowtie R_2$$

## Zerlegung von Relationen (3)

- Beispiel 1:

FBSTUDENT (MATNR, NAME, FBNR, FBADR)

mit MATNR  $\rightarrow$  NAME, FBNR, FBADR

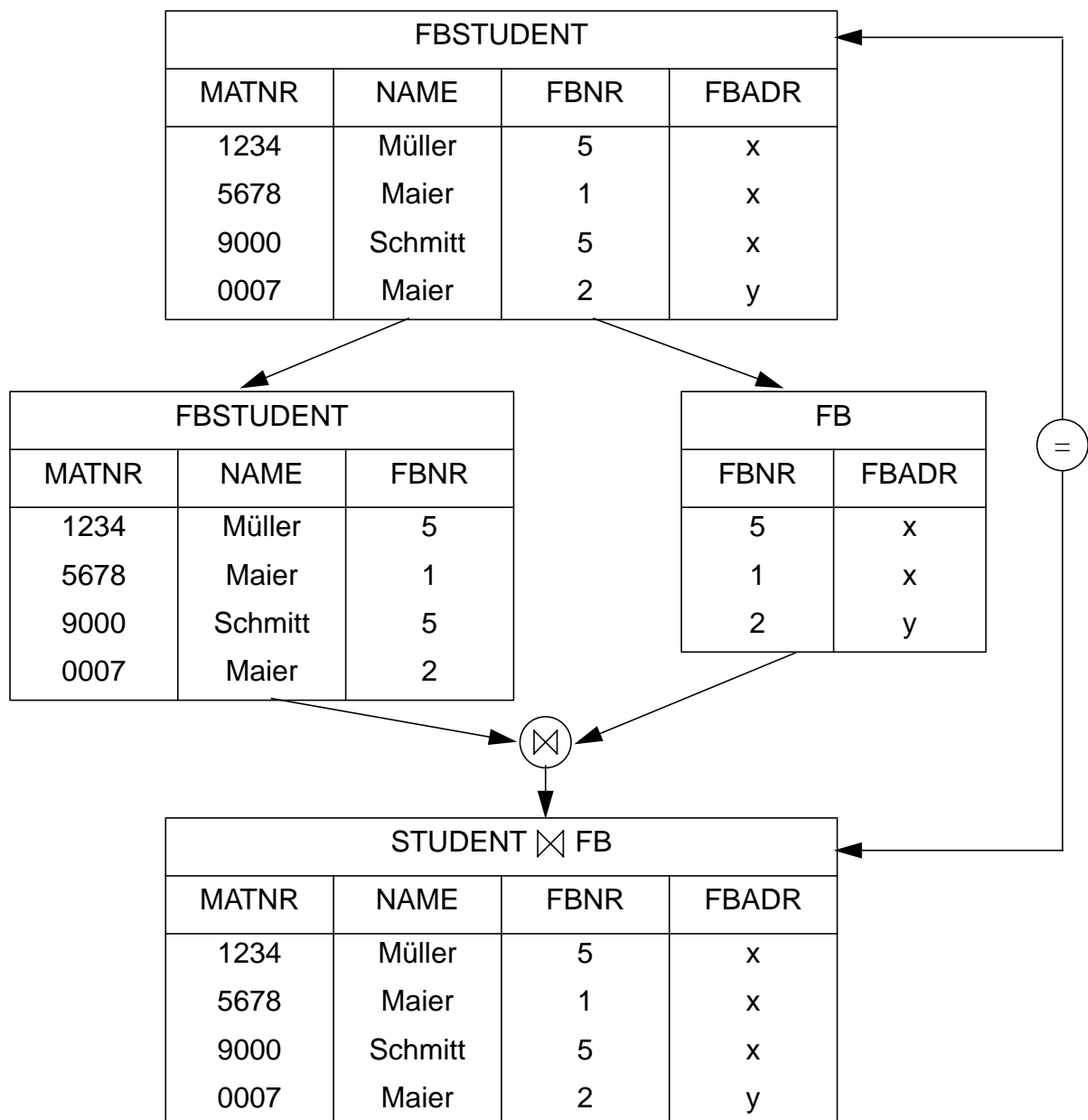
FBNR  $\rightarrow$  FBADR

STUDENT :=  $\Pi_{\text{MATNR, NAME, FBNR}}$  (FBSTUDENT)

FB :=  $\Pi_{\text{FBNR, FBADR}}$  (FBSTUDENT)

$\rightarrow$  FBSTUDENT = STUDENT  $\bowtie$  FB

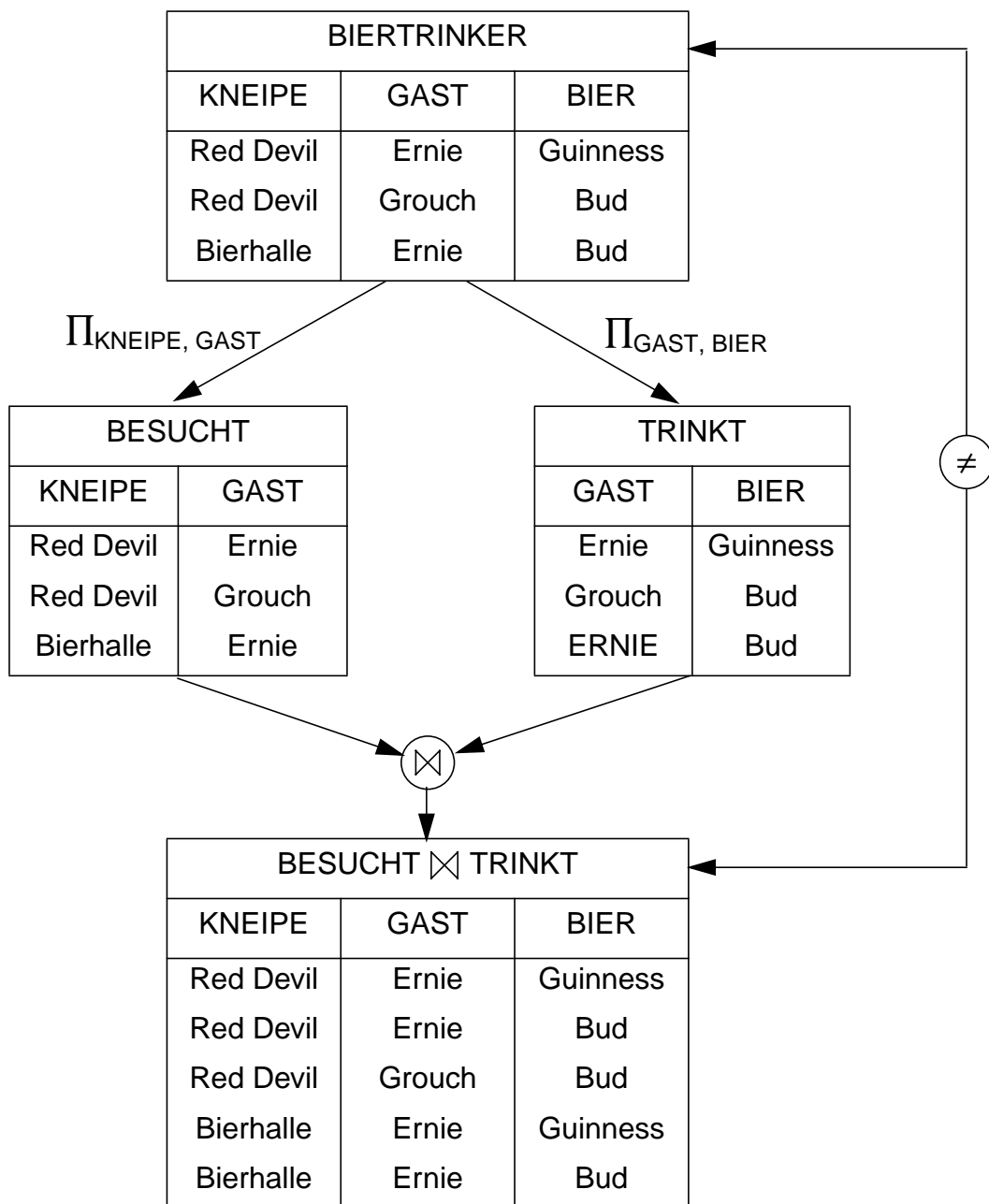
- Eine verlustlose Zerlegung von FBSTUDENT





## Zerlegung von Relationen (4)

- **Beispiel 2:** Verlustbehaftete Zerlegung von BIERTRINKER



- **Warum?**

KNEIPE, GAST  $\rightarrow$  BIER ist die einzige nicht-triviale FA von BIERTRINKER (Schlüssel wurde aufgeteilt!)

- Welche FA's würden eine verlustlose Zerlegung zulassen?



## Zerlegung von Relationen (6)

- **Andere Formulierung der Bedingung für Verlustlosigkeit**

- $R = X \cup Y \cup Z$

- $R_1 = X \cup Y$

- $R_2 = X \cup Z$

- $Y \cap Z = \emptyset$

- Einsatz des Algorithmus CLOSURE (Attributhülle)

1.  $Y \subseteq \text{CLOSURE}(X, F_R)$  oder

2.  $Z \subseteq \text{CLOSURE}(X, F_R)$

# Abhängigkeitsbewahrung

- Zerlegung von  $R$  mit den zugehörigen FA's  $F_R$

sollte so erfolgen, daß

- $R$  ist zerlegt in  $R_1, \dots, R_n$
- $F_R \equiv (F_{R_1} \cup \dots \cup F_{R_n})$  bzw.  $F_R^+ = (F_{R_1} \cup \dots \cup F_{R_n})^+$

➔ Überprüfung aller FA's sollte lokal auf den  $R_i$  erfolgen können. Eine solche abhängigkeitsbewahrende Zerlegung nennt man auch eine hüllentreue Dekomposition

- Beispiel für Abhängigkeitsverlust

PLZverzeichnis (Straße, Ort, BLand, PLZ)

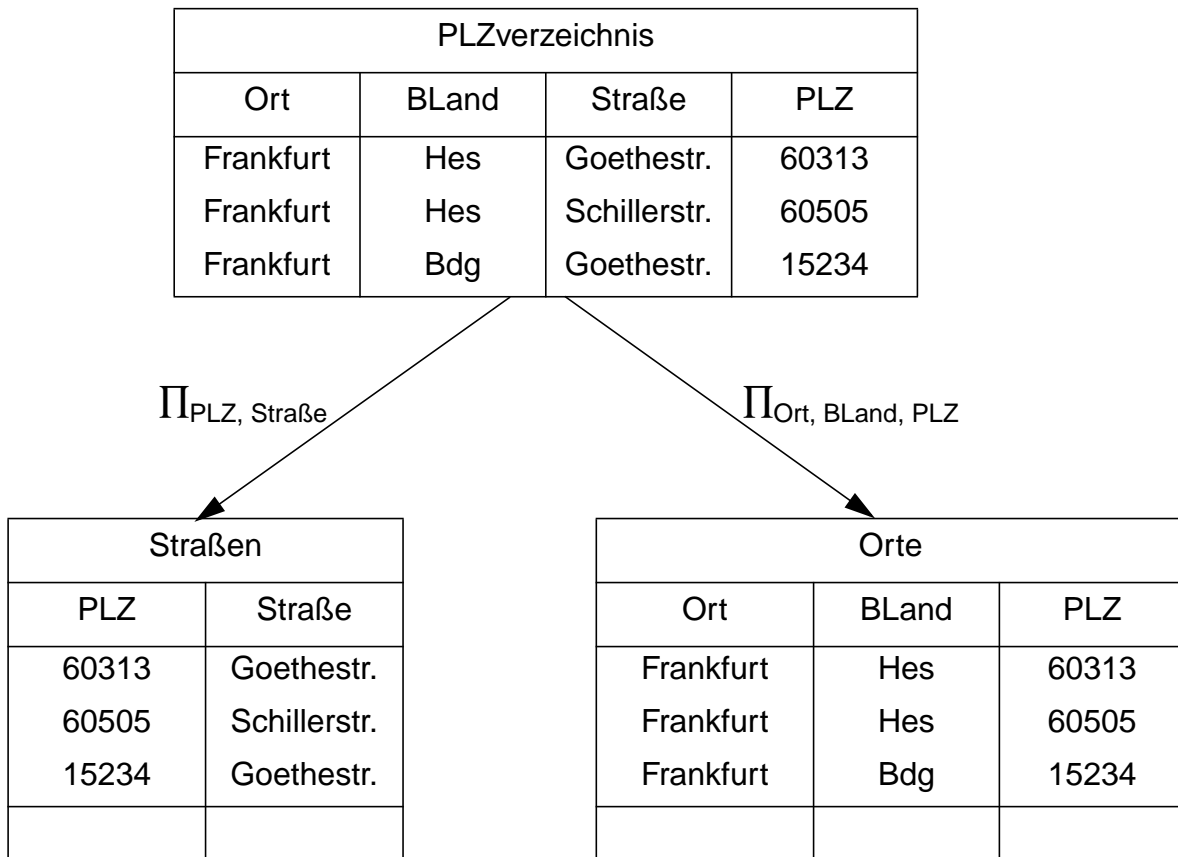
- Orte werden durch ihren Namen (Ort) und das Bundesland (BLand) eindeutig identifiziert
- Innerhalb einer Straße ändert sich die PLZ nicht
- PLZ-Gebiete gehen nicht über Ortsgrenzen und Orte nicht über BLand-Grenzen hinweg

PLZ → ORT, BLand

Straße, Ort, BLand → PLZ

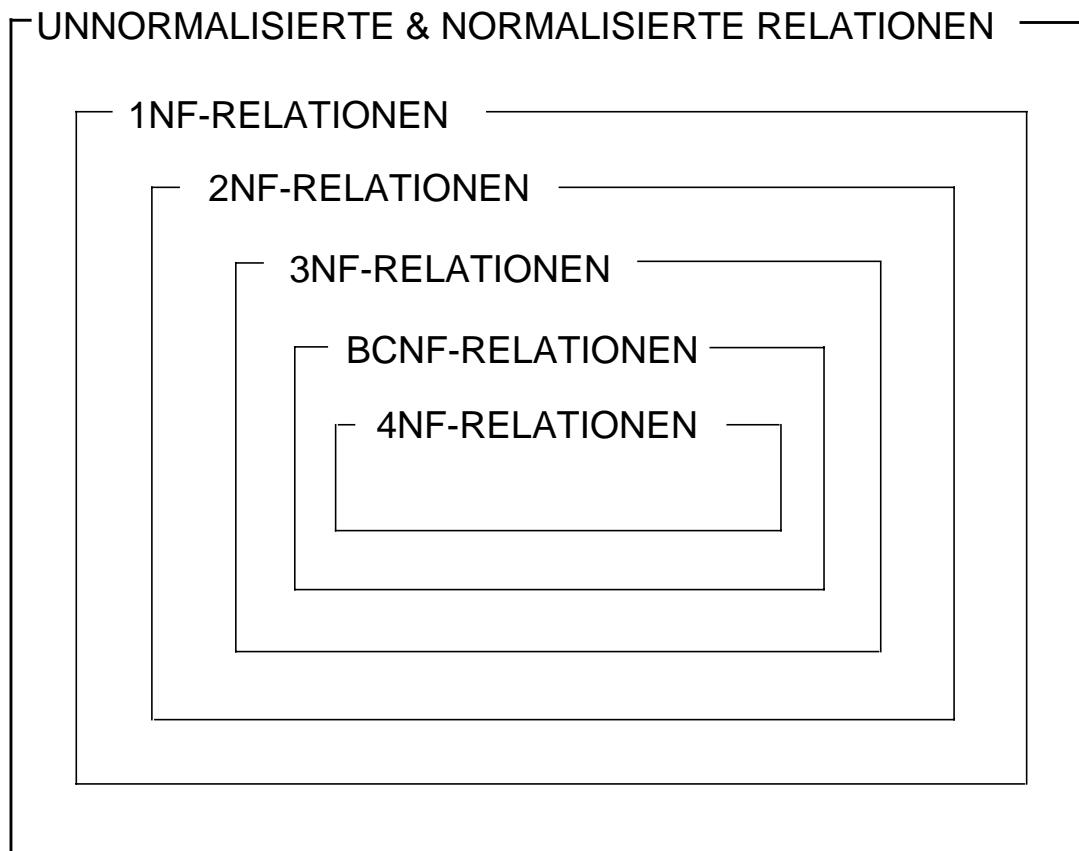
## Abhängigkeitsbewahrung (2)

- Beispiel



- Die FA Straße, Ort, BLand  $\rightarrow$  PLZ ist im zerlegten Schema nicht mehr enthalten
- Einfügen eines Eintrags:  
„Frankfurt, Bdg, Goethestr., 15235“  
führt auf Verletzung dieser FA

# Normalisierung von Relationen



- **Zerlegung eines Relationenschemas  $R$  in höhere Normalformen**
  - Beseitigung von Anomalien bei Änderungsoperationen
  - fortgesetzte Anwendung der Projektion im Zerlegungsprozeß
  - bessere „Lesbarkeit“ der aus  $R$  gewonnenen Relationen
  - Erhaltung aller nicht-redundanter Funktionalabhängigkeiten von  $R$   
( $\rightarrow$  sie bestimmen den Informationsgehalt von  $R$ )
  - Verlustlosigkeit der Zerlegung ist in alle Normalformen gewährleistet
  - Abhängigkeitserhaltung kann nur bei Zerlegungen bis zur 3NF garantiert werden

# Normalisierung von Relationen

- **Unnormalisierte Relation: Non-First Normal-Form (NF<sup>2</sup>)**

Relation enthält „Attribute“, die wiederum Relationen sind

PRÜFUNGSGESCHEHEN					
	( <u>PNR</u> ,	PNAME,	FACH,	STUDENT	(MATNR, NAME, ...))
1	Härder	DBS		1234	Müller
				5678	Maier
				9000	Schmitt
2	Schock	FA		5678	Maier
				007	Coy

## ➔ Darstellung von komplexen Objekten (Hierarchische Sichten)

- **VORTEILE:** Clusterbildung,  
Effiziente Verarbeitung in einem hierarchisch strukturierten Objekt längs der Vorzugsrichtung
- **NACHTEILE:** Unsymmetrie (nur eine Richtung der Beziehung),  
implizite Darstellung von Information,  
Redundanzen bei (n:m)-Beziehungen,  
Anomalien bei Aktualisierung,  
Definiertheit des Vaters

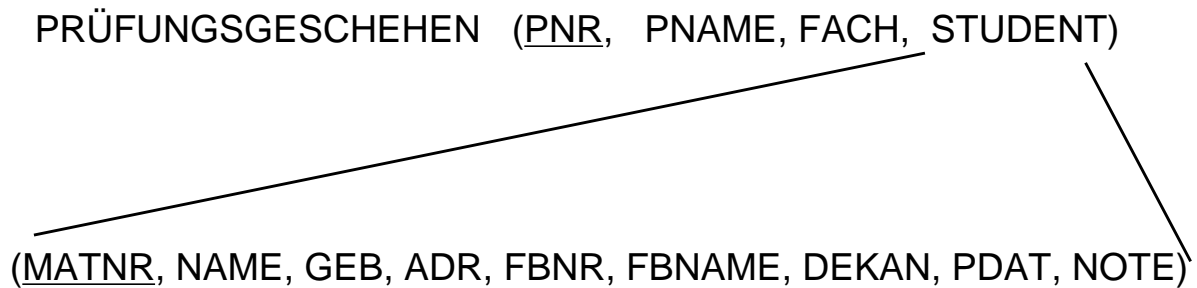
- **Normalisierung:**

- „Herunterkopieren“ von Werten führt hohen Grad an Redundanz ein

### ➔ Zerlegung von Relationen

- aber: Erhaltung ihres Informationsgehaltes

## Unnormalisierte Relation



STUDENT = Wiederholungsgruppe mit 9 einfachen Attributen (untergeordnete Relation)

- **Normalisierung (Überführung in 1NF):**

1. Starte mit der übergeordneten Relation (Vaterrelation)
2. Nimm ihren Primärschlüssel und erweitere jede unmittelbar untergeordnete Relation damit zu einer selbständigen Relation
3. Streiche alle nicht-einfachen Attribute (untergeordnete Relationen) aus der Vaterrelation
4. Wiederhole diesen Prozeß ggf. rekursiv

- **REGELN:**

- Nicht-einfache Attribute bilden neue Relationen
- Primärschlüssel der übergeordneten wird an untergeordnete Relation angehängt ('copy down the key')

- **Relationenschema in 1NF:**

PRÜFER (PNR, PNAME, FACH)

PRÜFUNG (PNR, MATNR, NAME, GEB, ADR, FBNR, FBNAME, DEKAN, PDAT, NOTE)



# Überführung in 2NF

- **Beobachtung**

- 1NF verursacht immer noch viele Änderungsanomalien, da verschiedene Entity-Mengen in einer Relation gespeichert werden können bzw. aufgrund von Redundanz innerhalb einer Relation (Beispiel: PRÜFUNG)
- 2NF vermeidet einige der Anomalien dadurch, indem nicht voll funktional (partiell) abhängige Attribute eliminiert werden

↳ **Separierung verschiedener Entity-Mengen in eigene Relationen**

- **Definition:**

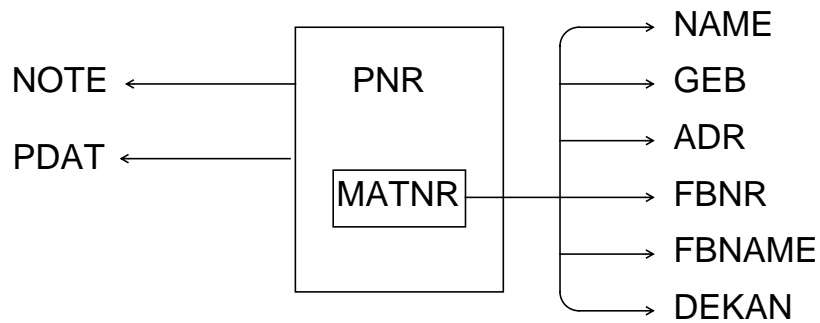
Ein **Primärattribut** (Schlüsselattribut) eines Relationenschemas ist ein Attribut, das zu mindestens einem Schlüsselkandidaten des Schemas gehört.

Ein Relationenschema  $R$  ist in **2NF**, wenn es in 1NF ist und jedes Nicht-Primärattribut von  $R$  voll funktional von jedem Schlüsselkandidaten in  $R$  abhängt.

- **Überführung in 2NF**

1. Bestimme funktionale Abhängigkeiten zwischen Nicht-Primärattributen und Schlüsselkandidaten
2. Eliminiere partiell abhängige Attribute und fasse sie in eigener Relation zusammen (unter Hinzunahme der zugehörigen Primärattribute)

## Voll funktionale Abhängigkeiten in PRÜFUNG



## Relationenschema in 2NF

### PRÜFUNG'

<u>PNR</u>	<u>MATNR</u>	PDAT	NOTE
1234	123 766	221001	4
1234	654 711	140200	3
3678	196 481	210999	2
3678	123 766	020301	4
8223	226 302	120701	1

### PRÜFER

<u>PNR</u>	PNAME	FACH
1234	Schock	FA
3678	Härder	DBS
8223	Franke	FM

### STUDENT'

<u>MATNR</u>	NAME	GEB	ADR	FBNR	FBNAME	DEKAN
123 766	Coy	050578	XX	FB1	Mathematik	Freden
654 711	Abel	211176	XY	FB9	Informatik	Hagen
196 481	Maier	010179	YX	FB9	Informatik	Hagen
226 302	Schulz	310778	YY	FB1	Mathematik	Freden

# Überführung in 3NF

- **Beobachtung**

Änderungsanomalien in 2NF sind immer noch möglich aufgrund von transitiven Abhängigkeiten.

Beispiel:

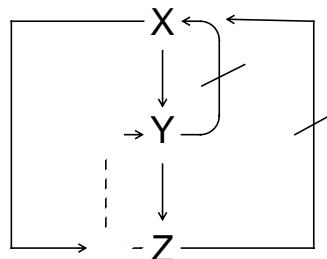
Vermischung von Fachbereichs- und Studentendaten in Student'

- **Definition:**

Eine Attributmenge  $Z$  von Relationenschema  $R$  ist transitiv abhängig von einer Attributmenge  $X$  in  $R$ , wenn gilt:

- $X$  und  $Z$  sind disjunkt
- Es existiert eine Attributmenge  $Y$  in  $R$ , so daß gilt:

$$X \longrightarrow Y, Y \longrightarrow Z, Y \not\rightarrow X, Z \not\subseteq Y$$

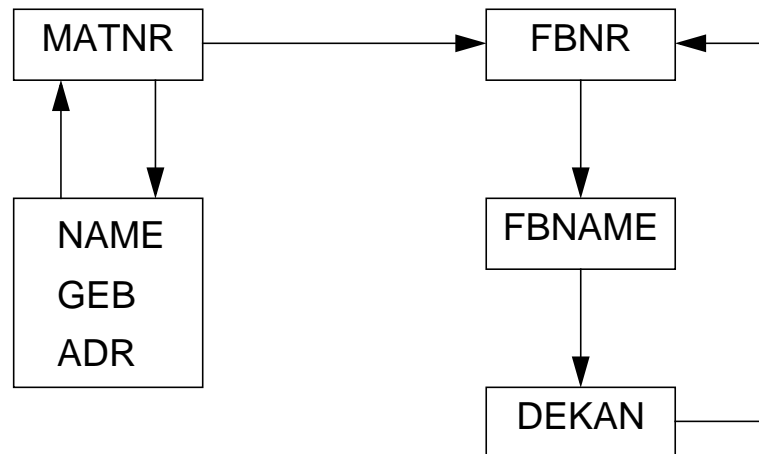


$Z \rightarrow Y$  zulässig

strikte Transitivität:  $Z \not\rightarrow Y$

Ein Relationenschema  $R$  befindet sich in **3NF**, wenn es sich in 2NF befindet und jedes Nicht-Primärattribut von  $R$  von keinem Schlüsselkandidaten von  $R$  transitiv abhängig ist.

## Funktionale Abhängigkeiten in 'STUDENT'



## Relationenschema in 3NF

### PRÜFUNG'

<u>PNR</u>	<u>MATNR</u>	PDAT	NOTE
1234	123 766	221001	4
1234	654 711	140200	3
3678	196 481	210999	2
3678	123 766	020301	4
8223	226 302	120701	1

### PRÜFER

<u>PNR</u>	PNAME	FACH
1234	Schock	FA
3678	Härder	DBS
8223	Franke	FM

### 'STUDENT'

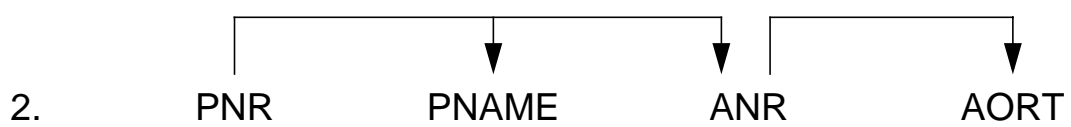
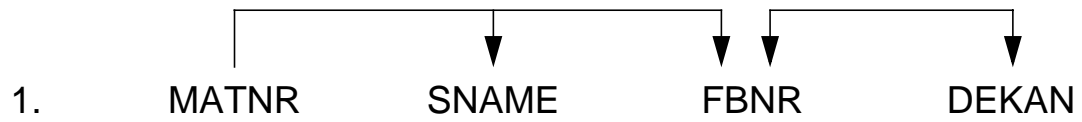
<u>MATNR</u>	NAME	GEB	ADR	FBNR
123 766	Coy	050578	XX	FB1
654 711	Abel	211176	XY	FB9
196 481	Maier	010179	YX	FB9
226 302	Schulz	310778	YY	FB1

### FACHBEREICH

<u>FBNR</u>	FBNAME	DEKAN
FB1	Mathematik	Freeden
FB9	Informatik	Hagen
FB2	Physik	Jodl

## 3NF-Relationen – Beispiel

- Eliminierung von transitiven Abhängigkeiten



- Zerlegung in 3NF-Relationen:

## Boyce/Codd-Normalform (BCNF)

- Definition der 3NF hat gewisse Schwächen bei Relationen mit mehreren Schlüsselkandidaten, wenn die Schlüsselkandidaten
  - zusammengesetzt sind und
  - sich überlappen
- Beispiel:

PRÜFUNG (PNR, MATNR, FACH, NOTE)  
PRIMARY KEY (PNR, MATNR)  
UNIQUE (FACH, MATNR)

- es bestehe eine (1:1)-Beziehung zwischen PNR und FACH
- einziges Nicht-Primärattribut: NOTE → PRÜFUNG ist in 3NF

PRÜFUNG	(PNR,	MATNR,	FACH,	NOTE)
	4	4711	BS	1
	4	1007	BS	2
	4	1234	BS	2
	5	4711	RO	3

→ Änderungsanomalien z. B. bei FACH

- ZIEL:  
Ausschluß/Beseitigung der Anomalien in den Primärattributen

## BCNF(2)

- **Definition:**

Ein Attribut (oder eine Gruppe von Attributen), von dem andere voll funktional abhängen, heißt Determinant.

Welches sind die Determinanten in PRÜFUNG?

- **Definition:**

Ein Relationenschema  $R$  ist in **BCNF**, wenn jeder Determinant ein Schlüsselkandidat von  $R$  ist.

- **Formale Definition:**

Ein Relationenschema ist in **BCNF**, falls gilt: Wenn eine Sammlung von Attributen  $Y$  (voll funktional) abhängt von einer disjunkten Sammlung von Attributen  $X$ , dann hängt jede andere Sammlung von Attributen  $Z$  auch von  $X$  (voll funktional) ab.

D. h. für alle  $X, Y, Z$  mit  $X$  und  $Y$  disjunkt gilt:

$$X \rightarrow Y \text{ impliziert } X \rightarrow Z$$

- Zerlegung von Prüfung

PRÜF1 (PNR, MATNR, NOTE)	oder	PRÜF2 (MATNR, FACH, NOTE)
FBEZ (PNR, FACH)		FBEZ (PNR, FACH)

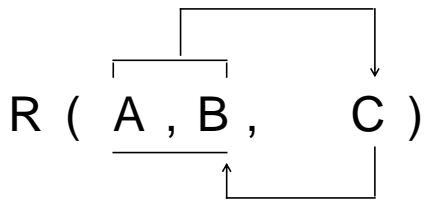
- Beide Zerlegungen führen auf BCNF-Relationen

- Änderungsanomalie ist verschwunden
- Alle funktionalen Abhängigkeiten sind erhalten

- Wann ergeben sich nach BCNF- und 3NF-Definition gleiche Zerlegungen?

## BCNF(3)

- Sind BCNF-Zerlegungen immer sinnvoll?



ist in 3NF, weil B ja  
Primärattribut ist!

Beispiel: STUDENT, FACH → PRÜFER  
PRÜFER → FACH

SFP (STUDENT FACH PRÜFER)

Sloppy	DBS	Härder
Hazy	DBS	Ritter
Sloppy	BS	Nehmer

- Jeder Prüfer prüft nur ein Fach  
(aber ein Fach kann von mehreren geprüft werden)
  - Jeder Student legt in einem bestimmten Fach nur eine Prüfung ab
- Wie sieht die BCNF-Zerlegung aus?

- **Neue Probleme**

- STUDENT, FACH → PRÜFER ist nun „extern“

➔ Konsistenzprüfung?

- BCNF ist hier zu streng, um bei der Zerlegung alle funktionalen Abhängigkeiten zu bewahren (*key breaking dependency*)



## Mehrwertige Abhängigkeiten (MWA)

- Eine FA bestimmt jeweils (höchstens) ein Wert des abhängigen Attributes
- MWA's sind Generalisierungen von FA's
  - Sie bestimmen jeweils eine Menge von Werten des abhängigen Attributes
  - Sie entstehen durch zwei (oder mehr) unabhängige Attribute im Schlüssel einer Relation (all-key relation): z.B. Fähigkeiten : Kinder
- Beispiel:

<u>PNR</u>	<u>FÄHIGKEIT</u>	<u>KIND</u>
123	Englisch	Nadine
123	Englisch	Philip
123	Englisch	Tobias
123	Programmieren	Nadine
123	Programmieren	Philip
123	Programmieren	Tobias

➔ **Änderungsanomalien** (obwohl in BCNF)

### • Definition:

X, Y, Z seien Attributmengen des Relationenschemas  $R$ .

Die **mehrwertige Abhängigkeit (MWA)**

$$X \twoheadrightarrow Y$$

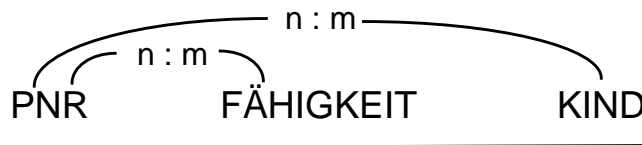
gilt in  $R$  genau dann, wenn die Menge der Y-Werte, die zu einem (X-Wert, Z-Wert)-Paar gehören, nur vom X-Wert bestimmt sind (d. h. unabhängig vom Z-Wert sind).

- MWA im Beispiel:  $PNR \twoheadrightarrow FÄHIGK.$ ,  $PNR \twoheadrightarrow KIND$
- $X \twoheadrightarrow Y$  impliziert  $X \twoheadrightarrow Z$   
Schreibweise:  $X \twoheadrightarrow Y \mid Z$ , z.B.  $PNR \twoheadrightarrow FÄHIGK. \mid KIND$
- Jede FA ist auch eine MWA.

## 4NF

- 4NF behandelt Probleme mit mehrwertigen Abhängigkeiten

- Schlüssel darf nicht 2 oder mehr **unabhängige mehrwertige** Fakten enthalten
- Beispiel



Zerlegung:



Andere Attribute sind erlaubt:



PNR	FÄHIGK.	PNR	KIND
123	Englisch	123	Nadine
123	Prog.	123	Philip
		123	Tobias

- Definition:**

Ein Relationenschema  $R$  ist in **4NF**, wenn es in BCNF ist und jede MWA in  $R$  eine FA ist.

- Überführung in 4NF**

Zerlege Relationenschema mit MWA  $X \twoheadrightarrow Y \mid Z$

in zwei neue Relationenschemata mit den Attributen  $X, Y$  bzw.  $X, Z$ .

## Abhängigkeit bei mehrwertigen Fakten

- Wenn Abhängigkeit besteht, muß sie durch die Wertekombinationen ausgedrückt werden
- **Beispiel**
  - (n:m)-Beziehung zwischen : PNR - PROJEKT, PNR - FÄHIGKEIT
  - zusätzliche (n:m)-Beziehung zwischen PROJEKT und FÄHIGKEIT, d. h., Projektmitarbeit erfordert bestimmte Fähigkeiten
- **Gültige Relation in 4NF: R (PNR, PROJEKT, FÄHIGKEIT)**

<u>PNR</u>	<u>PROJEKT</u>	<u>FÄHIGKEIT</u>
123	P1	Programmieren
123	P2	Programmieren
123	P2	Englisch

➔ Zerlegung von R in zwei Projektionen R1 (PNR, PROJEKT) und R2 (PNR, FÄHIGKEIT) führt zu 'Verlust' von Information, da Join-Bildung auf den Projektionen vorher nicht existente Tupel generieren kann (*connection trap*).

- **Gibt es andere verlustfreie Zerlegungen?**
  - Es gibt Relationen, die nicht verlustfrei in zwei, aber unter bestimmten Bedingungen verlustfrei in n ( $n > 2$ ) Projektionen (n-zerlegbar) zerlegt werden können
  - Zerlegung der Beispielrelation R in drei Projektionen R1 (PNR, PROJEKT), R2 (PNR, FÄHIGKEIT) sowie R3 (PROJEKT, FÄHIGKEIT) ist verlustfrei, d.h. Join zwischen diesen drei Projektionen erzeugt genau die Ausgangsrelation !
  - Ist eine solche Zerlegung in drei Projektionen sinnvoll ?

## Normalformenlehre nach E. F. Codd

**1NF:** Ein Relationenschema  $R$  ist in 1NF genau dann, wenn alle seine Wertebereiche nur atomare Werte besitzen.

**2NF:** Ein Relationenschema  $R$  ist in 2NF, wenn es in 1NF ist und jedes Nicht-Primärattribut von  $R$  voll funktional von jedem Schlüsselkandidaten von  $R$  abhängt.

**3NF:** Ein Relationenschema  $R$  ist in 3NF, wenn es in 2NF ist und jedes Nicht-Primärattribut von keinem Schlüsselkandidaten von  $R$  transitiv abhängig ist.

**3NF (BCNF):** Ein Relationenschema  $R$  ist in BCNF, falls gilt:

Wenn eine Sammlung von Attributen  $Y$  (voll funktional) abhängt von einer disjunkten Sammlung von Attributen  $X$ , dann hängt jede andere Sammlung von Attributen  $Z$  auch von  $X$  (voll funktional) ab.

Das heißt, für alle  $X, Y, Z$  mit  $X$  und  $Y$  disjunkt gilt:

$$X \rightarrow Y \text{ impliziert } X \rightarrow Z$$

- **Alternative Definition der BCNF:**

Ein normalisiertes Relationenschema  $R$  ist in 3NF (BCNF), wenn jeder Determinant in  $R$  ein Schlüsselkandidat von  $R$  ist.

**4NF:** Ein Relationenschema  $R$  ist in 4NF, wenn es in BCNF ist und jede MWA auf  $R$  eine FA ist.

# Entwurfstheorie für relationale Datenbanken

- **Erhebung der Miniwelt (Informationsbedarfsanalyse) liefert:**
  - Menge aller Attribute (*universelles Relationenschema*)
  - Menge  $F$  der funktionalen Abhängigkeiten zwischen Attributen
    - ↳ **Synthese-Verfahren erzeugt daraus relationales DB-Schema in 3NF**
- **Synthese erfordert u. a.**
  - alle bereits eingeführten Inferenzregeln (*Armstrong-Axiome*) zur Bestimmung der Schlüssel und zum Verstehen der logischen Implikationen (Elimination von Redundanz)
  - Verfahren wie CLOSURE und MEMBER (siehe Bestimmung funktionaler Abhängigkeiten)
  - Ableitung einer minimalen (kanonischen) Überdeckung von  $F$ , um ein redundanzfreies DB-Schema zu gewährleisten
- **Berechnung von  $F^+$** 
  - wozu?
  - $F^+$  wird nur als Hilfsbegriff für die Bestimmung der Äquivalenz von Familien funktionaler Abhängigkeiten benötigt
- **Definition: Äquivalenz von FA-Mengen**

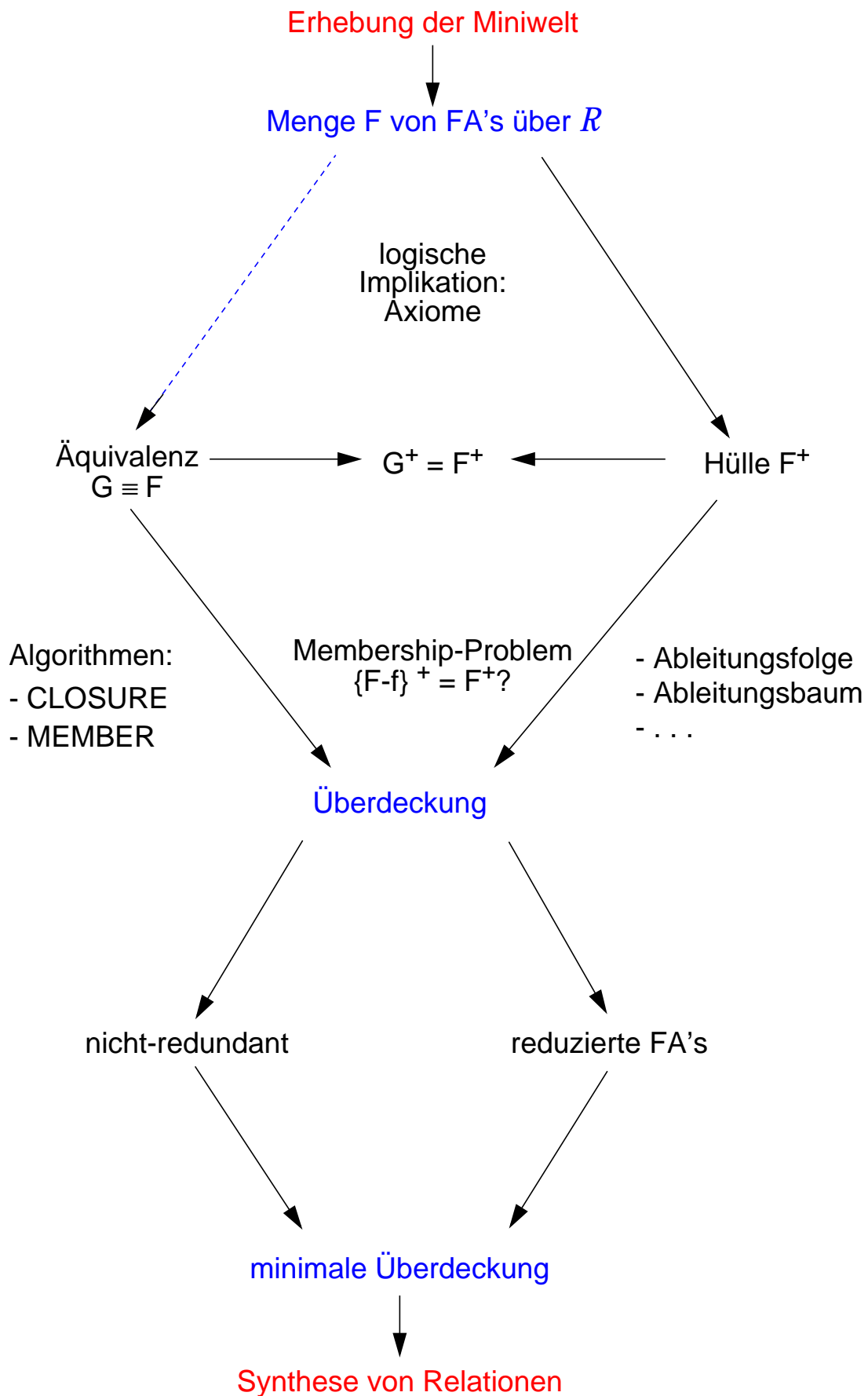
Seien  $F$  und  $G$  zwei Mengen von FA's über  $A$ .

$F$  und  $G$  sind äquivalent, wenn  $F^+ = G^+$ ,

d. h., wenn ihre Hüllen gleich sind.

$F$  heißt auch Überdeckung von  $G$  und umgekehrt.

# Entwurfstheorie für relationale Datenbanken – Überblick



## Entwurfstheorie für relationale Datenbanken (2)

- **Lösungsverfahren für das Membership-Problem:**

Gegeben  $F, X \rightarrow Y$ :

Ist  $X \rightarrow Y$  in  $F^+$  ?

Ist  $X \rightarrow Y$  in  $(F - \{X \rightarrow Y\})^+$  ?

### 1. Ableitungsfolge (Anwendung der Axiome)

Definition: Eine Folge  $P$  von FA's über  $R$  ist eine **Ableitungsfolge** auf  $F$ , wenn jede FA in  $P$

- 1) entweder aus  $F$  ist oder
- 2) aus vorangehenden FA's in  $P$  durch die Axiome A1 – A3 folgt.

- **Haupteigenschaft einer Ableitungsfolge:**

Wenn  $X \rightarrow Y$  durch  $P$  abgeleitet werden kann, dann ist  $X \rightarrow Y \in F^+$ .

# Entwurfstheorie für relationale Datenbanken (3)

## 2. Ableitungsbäume (graphische Methode)

Definition: **Ableitungsbaum**

- 1) Ein Knoten mit der Marke  $A$  ist ein Ableitungsbaum für  $F$ .
- 2) Wenn  $T$  ein Ableitungsbaum mit dem Knoten  $A$  als Blatt ist und  $B_1, B_2, \dots, B_m \rightarrow A \in F$  gilt, dann ist  $T'$  durch Anhängen von  $B_1, B_2, \dots, B_m$  auch ein Ableitungsbaum für  $F$ .

- **Haupteigenschaft eines Ableitungsbaumes:**

$Y$  sei nicht-leere Menge von Knoten eines Ableitungsbaumes  $AB$ .

$X$  sei die Menge aller Blätter von  $AB$ . Dann ist  $X \rightarrow Y \in F^+$  !



# Entwurfstheorie für relationale Datenbanken (4)

## 3. Ableitungsgraphen als Verallgemeinerung von Ableitungsbäumen

Definition: **Ableitungsgraph** (AG)

Sei  $F$  eine Menge von FA's über  $R$ . Ein AG ist ein **gerichteter azyklischer AG (GAAG)** mit Attributnamen von  $R$  als Marken, für den gilt:

- 1) Jede Menge unverbundener Knoten mit Namen aus  $R$  ist ein GAAG.
- 2) Sei  $H$  ein GAAG mit Knoten  $V_1, V_2, \dots, V_k$  und Marken  $A_1, A_2, \dots, A_k$  und sei  $A_1, A_2, \dots, A_k \rightarrow CZ$  eine FA in  $F$  ( $C$  Attribut,  $Z$  Attributmenge).  
Bilde  $H'$  unter Hinzufügen von Knoten  $U$  mit Marke  $C$  und Kanten  $(V_1, U), (V_2, U), \dots, (V_k, U)$  zu  $H$ .  $H'$  ist ein GAAG.
- 3) Nichts sonst ist ein GAAG.

### Haupteigenschaft eines Ableitungsgraphen

$H$  ist ein GAAG für  $X \rightarrow Y \in F^+$ , wenn

1.  $X$  die Menge der Anfangsknoten ist
1. Jedes Attribut von  $Y$  ein Knoten von  $H$  ist.

# Entwurfstheorie für relationale Datenbanken (5)

## Theorem

Gegeben seien eine Menge FA's  $F$  und eine FA  $X \rightarrow Y$ ; dann sind die folgenden Aussagen äquivalent:

1.  $F \models X \rightarrow Y$
2. Es gibt eine Ableitungsfolge auf  $F$  für  $X \rightarrow Y$
3. Es gibt einen GAAG über  $F$  für  $X \rightarrow Y$ .

# Entwurfstheorie für relationale Datenbanken (6)

- **Zerlegung**

Für eine gegebene Menge von Funktionalabhängigkeiten läßt sich eine äquivalente Menge von Funktionalabhängigkeiten mit jeweils einem Attribut auf der rechten Seite finden:

- Zerlegung mit Regel R5
- Wiedergewinnung der ursprünglichen Menge mit Regel R4  
( $\Rightarrow$  Äquivalenz)

- **Minimale Überdeckungen**

**Definition:** Eine Menge von Funktionalabhängigkeiten ***F ist minimal***, wenn gilt:

- 1) Jede rechte Seite von einer FA in F besteht aus einem Attribut.
- 2) Es gibt kein  $X \rightarrow A$  in F, so daß die Menge  $F - \{X \rightarrow A\}$  äquivalent zu F ist.
- 3) Es gibt kein  $X \rightarrow A$  in F und keine echte Untermenge Z von X, so daß  $F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$  äquivalent zu F ist.

# Entwurfstheorie für relationale Datenbanken (7)

- **Minimale Überdeckungen**

Für eine gegebene Menge von Funktionalabhängigkeiten läßt sich eine äquivalente Menge von Funktionalabhängigkeiten mit jeweils einem Attribut auf der rechten Seite finden

## Algorithmus MINCOVER:

Eingabe: Menge  $G$  von FA's mit jeweils minimaler linker Seite und einfacher rechter Seite

Ausgabe: minimale Überdeckung für  $G$

MINCOVER ( $G$ )

begin

$F := G$ ;

for jede FA  $X \rightarrow Y$  in  $G$  do

if MEMBER ( $F - \{X \rightarrow Y\}, X \rightarrow Y$ ) then

$F := F - \{X \rightarrow Y\}$ ;

return  $\{F\}$ ;

end

# Entwurfstheorie – Syntheseverfahren

- **Gegeben:**

- $A; F$  (erhoben in der Miniwelt)
- Modellannahme: universelles Relationenschema  $U$  enthält alle Attribute

- **Gesucht:**

Relationales DB-Schema  $RS$  mit folgenden Eigenschaften bezüglich  $U$ :

- 1) **Informationsgleichheit** (*lossless join decomposition*)
- 2) **Abhängigkeitsbewahrung** (*dependency preservation*)
- 3) **Redundanzminimierung**

zu 1: Jedes Attribut von  $U$  ist in mindestens einer Relation von  $RS$  enthalten.  
Die Zerlegung in mehrere Relationen ist verlustfrei

zu 2: Alle FA's der minimalen Überdeckung von  $F$  sind durch Schlüsselkandidaten in den Relationen von  $RS$  verkörpert

zu 3: Alle Relationen sind in 3NF; die Anzahl der Relationen ist minimal.

# Syntheseverfahren – Voraussetzungen

## 1. Eindeutigkeitsannahme

Wenn  $f: X \rightarrow Y$  und  $g: X \rightarrow Y$ , dann  $f \equiv g$

Beispiel:  $f_1: \text{PNR} \rightarrow \text{TELNR}$  (Angest. benutzt Telefon)

$f_2: \text{TELNR} \rightarrow \text{ANR}$  (Tel. wird abgerechnet über Abt.)

➔ abgeleitete FA

$f_{12}: \text{PNR} \rightarrow \text{ANR}$  („benutzt Telefon, das abgerechnet wird über“)

➔ i. allg. verschieden von der wohl erhobenen FA

$f_3: \text{PNR} \rightarrow \text{ANR}$  (Angest. gehört zu Abt.)

➔ Problem der Bedeutungstransitivität!

## 2. Darstellung nicht-funktionaler Beziehungen

$X \text{ — } Y$   $n : m$ , d. h.  $X \not\rightarrow Y$  und  $Y \not\rightarrow X$

➔  $XY \rightarrow \Theta$  mit  $\Theta$  „leeres Attribut“

# Synthese-Algorithmus

- **Eingabe:**  $A; F$
- **Ausgabe:** RS in 3NF mit minimaler Anzahl von Relationen

Schritt 1: **Ermittle eine minimale Überdeckung  $H$  für  $F$**   
( $\rightarrow$  MINCOVER( $F$ ))

Schritt 2: **Teile  $H$  in Partitionen mit gleichen linken Seiten auf**

Schritt 3: **Mische äquivalente Schlüssel**  
( $\rightarrow$  Schlüsselkandidaten sollen derselben Relation zugeordnet werden)

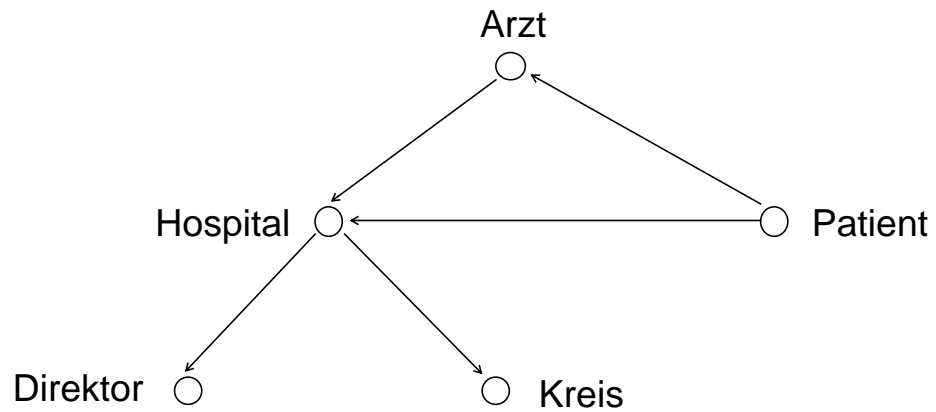
Schritt 4: **Eliminiere transitive Abhängigkeiten ( $\rightarrow H'$ )**, die durch Schritt 3 eingeführt wurden  
( $\rightarrow$  innerhalb von Schlüsselattributen)

Schritt 5: **Konstruiere für jede Partition von  $H'$  eine Relation**  
( $\rightarrow$  jede Attributmenge auf der linken Seite einer FA ist ein Schlüsselkandidat)

# Anwendung des Syntheseverfahrens

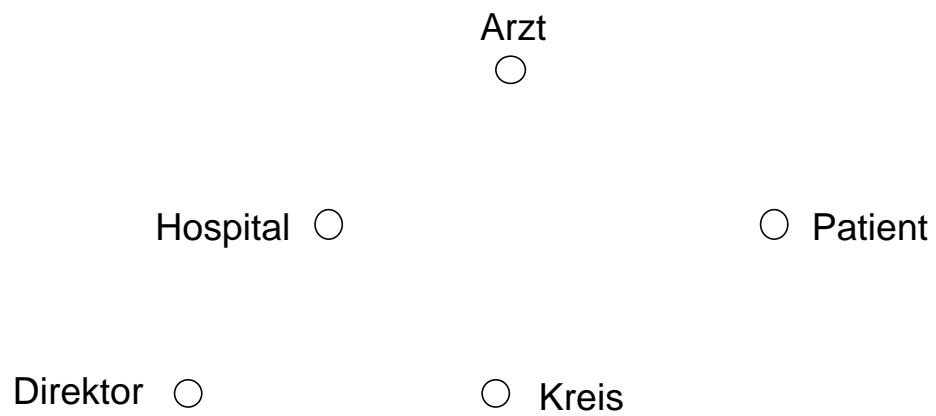
- **Beispiel 1:**

**F:**



**Wie sieht  $F^+$  aus?**

**H:**



Schritt 1:  $H =$

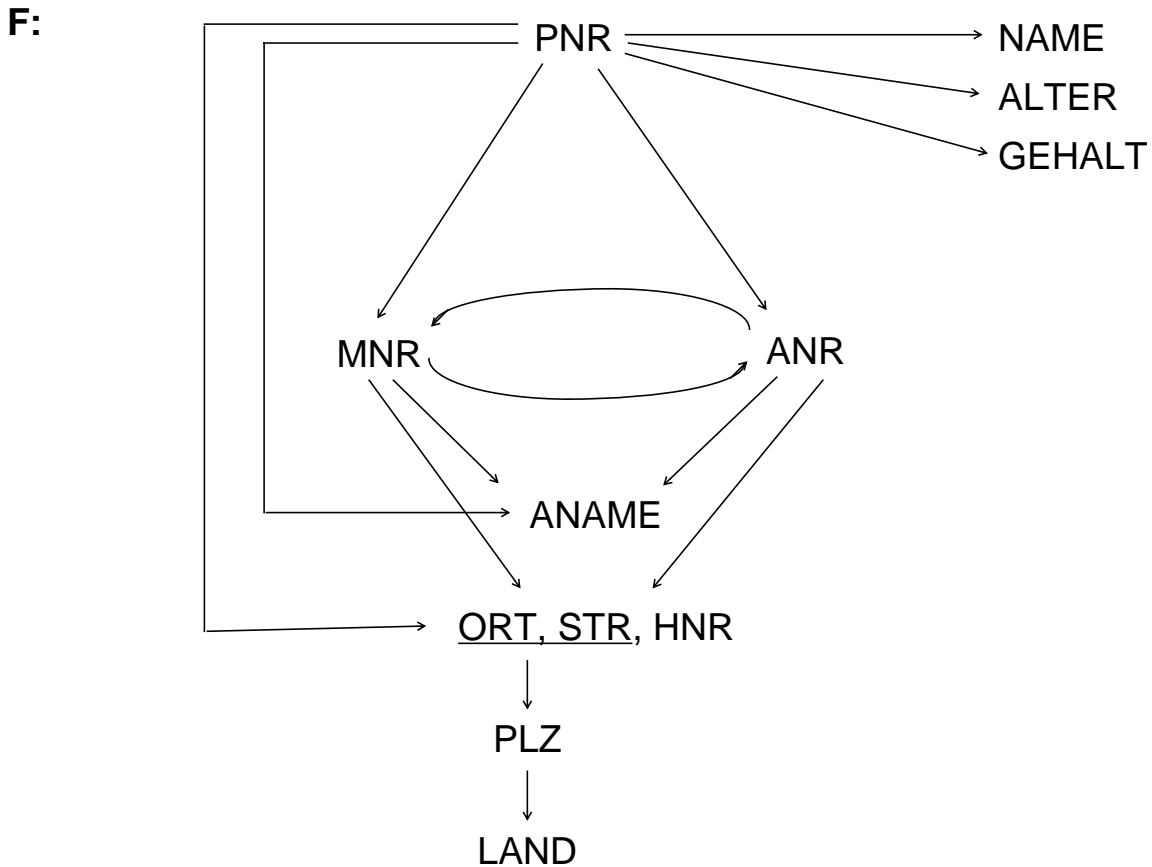
Schritt 2:  $g_1$   
 $g_2$   
 $g_3$

Schritt 5:  $R_1$   
 $R_2$   
 $R_3$



## Anwendung des Syntheseverfahrens (2)

- Beispiel 2:



- |                       |                       |                    |
|-----------------------|-----------------------|--------------------|
| f1: PNR → NAME        | f8: MNR → ANR         | f11: ANR → MNR     |
| f2: PNR → ALTER       | f9: MNR → ORT,STR,HNR | f12: ANR → ANAME   |
| f3: PNR → GEHALT      | f10: MNR → ANAME      | f13: ANR →         |
| f4: PNR → ANR         |                       | HNR                |
| f5: PNR → MNR         |                       | f14: ORT,STR → PLZ |
| f6: PNR → ORT,STR,HNR |                       | f15: PLZ → LAND    |
| f7: PNR → ANAME       |                       |                    |

- Auswahl einer geeigneten Überdeckung: semantische Kriterien!

## Anwendung des Syntheseverfahrens (3)

- **Beispiel 2:**

Wieviele minimale Überdeckungen  $H$  existieren?

1. Ermittle minimale Überdeckung
2. Partitioniere
3. Mische äquivalente Schlüssel
4. Eliminiere transitive Abhängigkeiten
5. Konstruiere Relationen

## Anwendung des Syntheseverfahrens (4)

- **Eine Lösung:**

R1 (PNR , NAME, ALTER, GEHALT, MNR)

R2 (MNR , ANR, ANAME, ORT, STR, HNR)

R3 (ORT, STR, PLZ)

R4 (PLZ, LAND)

- **Fragen:**

**1. Wie häufig treten in**

ORT, STR

Wiederholungen auf?

**2. Ist die Zerlegung von**

ORT, STR → PLZ → LAND

in R3 und R4 **sinnvoll?**

- Änderungshäufigkeit?
- Aufsuchen der Adresse (Verbundoperation) !
- Ist ORT, STR oder PLZ in diesem Kontext ein Entity?  
(als Kandidat für eigene Relation in 3NF)

➔ besser R2 in 2NF !

**3. Stabilität von MNR?**

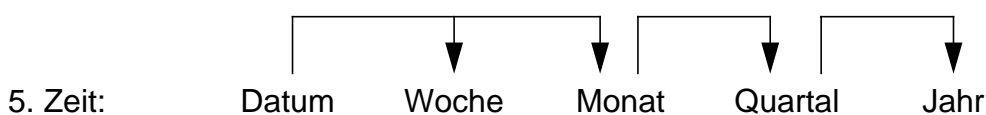
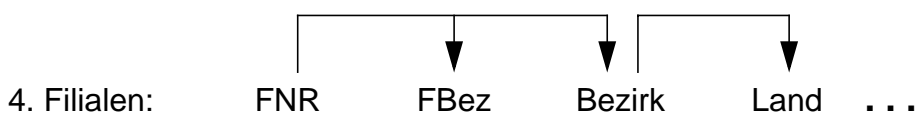
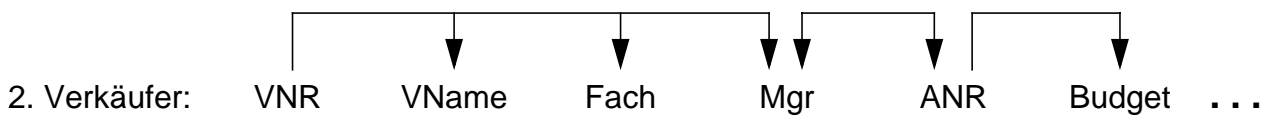
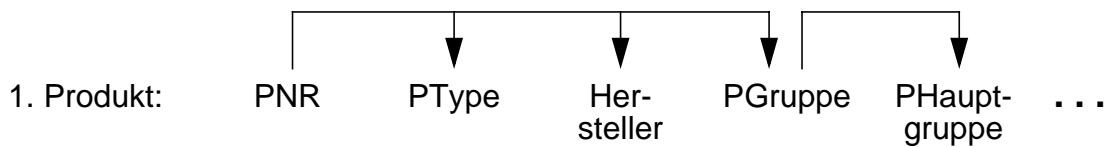
- Änderungshäufigkeit von ANR und MNR!

➔ R1 (PNR , NAME, ALTER, GEHALT, ANR)

R2 (ANR , MNR, ANAME, ORT, STR, HNR, PLZ, LAND)

# Schemasynthese bei Data Warehouses

- Funktionale Abhängigkeiten



# Schemasynthese bei Data Warehouses (2)

- Stern-/Schneeflocken-Schema

# Entwurfstheorie – Zusammenfassung

- **Festlegung aller funktionalen Abhängigkeiten**
  - unterstützt präzises Denken beim Entwurf
  - erlaubt Integritätskontrollen durch das DBS
- **ZIEL: klare und natürliche Zuordnung von Objekt und Datenstruktur**
  - ➔ „wachsender Informationsgehalt“ mit zunehmender Normalisierung
  - ➔ durch einen Satztyp (Relation) wird nur ein Objekttyp beschrieben
- **Normalisierung von Relationen**
  - lokales Verfahren auf existierenden Datenstrukturen
  - schrittweise Eliminierung von Änderungsanomalien
  - übergreifende Maßnahmen zur DB-Schema-Integration
- **Synthese von Relationen**
  - globales Verfahren liefert 3NF-Relationen
  - ggf. Überprüfung von überlappenden Schlüsselkandidaten, mehrwertigen Abhängigkeiten und Join-Abhängigkeiten
    - ➔ BCNF-, 4NF- bzw. 5NF-Zerlegung
- **Weitere Probleme**
  - Definition aller relevanten FA's bei sehr vielen Attributen schwierig
  - Entwurfs-Algorithmen liefern i. allg. mehrere minimale Überdeckungen
  - Bei Überführung von 3NF in BCNF können FA's verlorengehen
- **Überarbeitung des DB-Schemas**
  - Stabilitätsgesichtspunkte/Änderungshäufigkeiten können schwächere Normalformen erzwingen
  - Berücksichtigung von Abstraktionskonzepten
    - ➔ Der Entwerfer, und nicht die Methode, bestimmt den Entwurf