

6. Normalisierung von Relationen

- Einführung
- Funktionale Abhängigkeiten
 - Bestimmung von Schlüsselkandidaten
 - Äquivalenzbeziehungen
- Zerlegung von Relationen zur Beseitigung von Anomalien
 - Korrektheitskriterien
- Normalisierung
 - 1NF
 - 2NF, partielle Abhängigkeiten
 - 3NF, transitive Abhängigkeiten
 - BCNF (Boyce-Codd-Normalform), Determinanten
- Probleme der Normalisierung

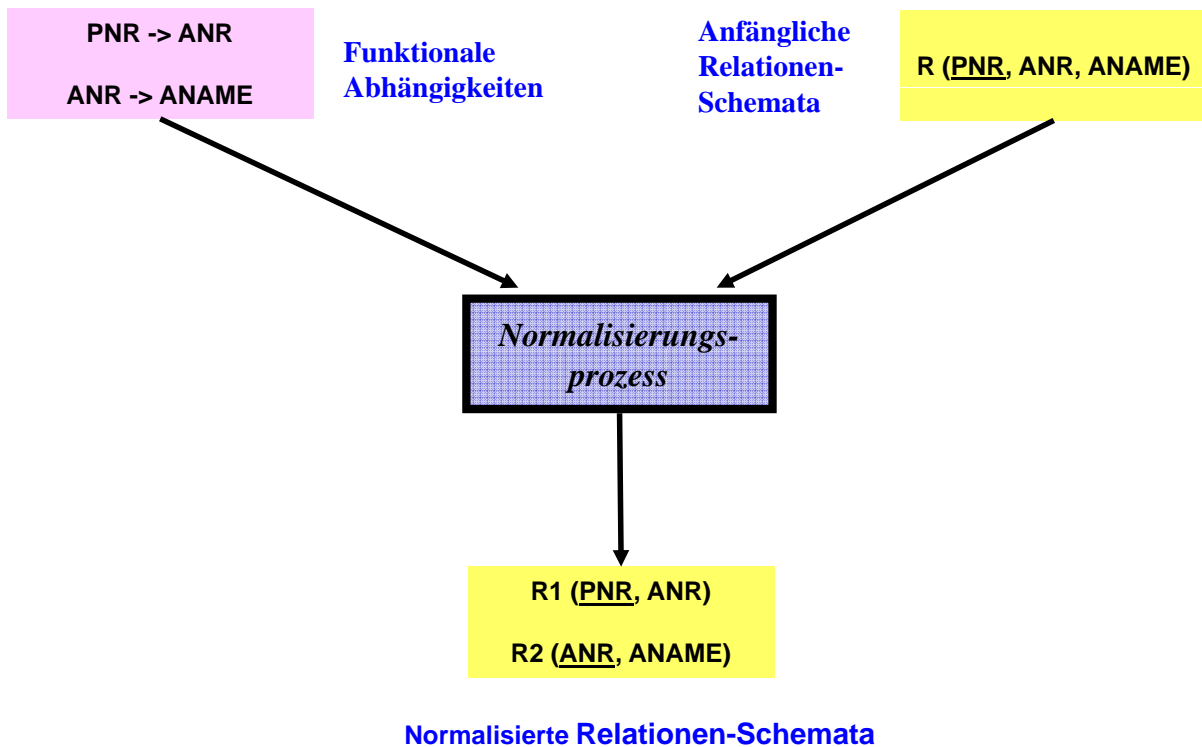


Einführung

- Ziel: Theoretische Grundlage für “gute” relationale DB-Schemas
- Normalisierung von Relationen: Verbesserung eines gegebenen Schema-Entwurfs
 - Teilweise Formalisierung von „Güte“ eines Schemas
 - Semiformales Verfahren zur Korrektur schlechter Schemas
- Merkmale eines schlechten DB-Schema-Entwurfs
 - 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 (Bsp.)



Definitionen und Begriffe

■ Konventionen

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

Funktionale Abhängigkeit

■ Def.: Funktionale Abhängigkeit (FA)

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.

$$\forall u \in R \forall v \in R (u[X] = v[X]) \Rightarrow (u[Y] = v[Y])$$

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

R			
A	B	C	D
a4	b2	c4	d3
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c3	d2
a3	b2	c4	d3

$A \rightarrow B$
 $C, D \rightarrow B$



Funktionale Abhängigkeiten (2)

■ Graphische Notation:



■ FA beschreiben semantische Integritätsbedingungen bezüglich der Attribute eines Rel.schemas, die jederzeit erfüllt sein müssen

■ Triviale FA:

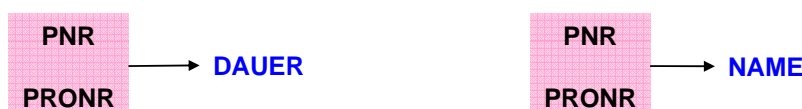
- $X \rightarrow Y$ und Y ist Teilmenge von X
- Spezialfall: $X \rightarrow X$

■ Definitionen: Voll funktionale vs. partielle Abhängigkeit

Sei $A_1, A_2, \dots, A_n \rightarrow B_1, B_2, \dots, B_m$

$B = \{B_1, B_2, \dots, B_m\}$ ist **voll funktional abhängig** von $A = \{A_1, A_2, \dots, A_n\}$, wenn B funktional abhängig von A ist, aber nicht funktional abhängig von einer echten Teilmenge von A ist.

$A \rightarrow B$ ist eine **partielle Abhängigkeit**, wenn ein Attribut A_i in A existiert, so dass $(A - \{A_i\}) \rightarrow B$ gilt.



FA und Schlüsselkandidaten

- X ist **Schlüsselkandidat** von \mathcal{R} , wenn
 - für alle Y aus \mathcal{R} : $X \rightarrow Y$
 - keine echte Teilmenge von X bestimmt funktional alle anderen Attribute Y aus \mathcal{R} (**Minimalität**)
- Kenntnis aller FA ermöglicht Bestimmung der Schlüsselkandidaten
 - Beispiel: Attribute A, B, C, D mit
 - B \rightarrow A,
 - B \rightarrow D,
 - C \rightarrow B



Funktionale Abhängigkeiten : Beispiel

- Beispieltabelle *Film*

Titel	Jahr	Dauer	FilmTyp	StudioName	StarName
Star Wars	1977	124	Farbe	Fox	Carrie Fisher
Star Wars	1977	124	Farbe	Fox	Mark Hamill
Star Wars	1977	124	Farbe	Fox	Harrison Ford
Good-Bye Lenin!	2003	121	Farbe	WDR	Daniel Brühl
Troja	2004	156	Farbe	Warner Bros	Brad Pitt
Troja	1956	118	SW	Warner Bros	Stanley Baker

- Funktionale Abhängigkeiten:

- Schlüsselkandidat:



Funktionale Abhängigkeiten: Äquivalenzbeziehungen

■ Splitten / Kombinieren von FA

Eine FA $A_1, A_2, \dots, A_n \rightarrow B_1, B_2, \dots, B_m$

ist äquivalent zu m FA $A_1, A_2, \dots, A_n \rightarrow B_1$
 \dots
 $A_1, A_2, \dots, A_n \rightarrow B_m$

■ Weitere Äquivalenzbeziehungen (Regeln zur Ableitung neuer aus gegebenen FA; Armstrong-Axiome)

- Reflexivität: wenn $X \subseteq Y$ dann $Y \rightarrow X$ (triviale FA)
- Komplementierung: wenn $X \rightarrow Y$ dann $XZ \rightarrow YZ$
- Transitivität: wenn $X \rightarrow Y, Y \rightarrow Z$ dann $X \rightarrow Z$



„Schlechte“ Relationenschemata

ProfVorl						
PersNr	Name	Fach	Raum	VorlNr	Titel	SWS
3678	Rahm	DBS	356	5041	DBS1	3
3678	Rahm	DBS	356	5049	DBS2	3
3678	Rahm	DBS	356	4052	IDBS	4
...
1234	Brewka	KI	152	5259	Wissensrepräsentation	2
2137	Meyer	TI	17	4630	Informationstheorie	4

■ Update-Anomalien

- Umzug von Raum 356 in Raum 338. Was passiert?

■ Einfüge-Anomalien

- Neuer Prof ohne Vorlesungen?

■ Löschanomalien

- Letzte Vorlesung eines Profs wird gelöscht? Was passiert?



Zerlegung (Dekomposition) von Relationen

■ Zwei Korrektheitskriterien für die Zerlegung von Relationenschemata:

1. **Verlustlosigkeit:** Die in der ursprünglichen Relationenausprägung R des Schemas \mathcal{R} enthaltenen Informationen müssen aus den Ausprägungen R_1, \dots, R_n der neuen Relationenschemata $\mathcal{R}_1, \dots, \mathcal{R}_n$ rekonstruierbar sein.
 - erfordert vollständige Aufteilung der Attributmengen, z.B. $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$

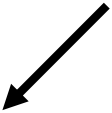
$$R_1 := \Pi_{\mathcal{R}_1}(R),$$

$$R_2 := \Pi_{\mathcal{R}_2}(R)$$
 - Verlustfreiheit verlangt, dass für jede mögliche (gültige) Ausprägung R von \mathcal{R} gilt:

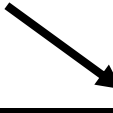
$$R = R_1 \bowtie R_2$$
2. **Abhängigkeitserhaltung:** Die für \mathcal{R} geltenden funktionalen Abhängigkeiten müssen auf die Schemata $\mathcal{R}_1, \dots, \mathcal{R}_n$ übertragbar sein.

Beispiel: Zerlegung der Filmtabelle

Titel	Jahr	Dauer	FilmTyp	StudioName	StarName
Star Wars	1977	124	Farbe	Fox	Carrie Fisher
Star Wars	1977	124	Farbe	Fox	Mark Hamill
Star Wars	1977	124	Farbe	Fox	Harrison Ford
Good-Bye Lenin!	2003	121	Farbe	WDR	Daniel Brühl
Troja	2004	156	Farbe	Warner Bros	Brad Pitt
Troja	1956	118	SW	Warner Bros	Stanley Baker



Titel	Jahr	Dauer	FilmTyp	StudioName
Star Wars	1977	124	Farbe	Fox
Good-Bye Lenin!	2003	121	Farbe	WDR
Troja	2004	156	Farbe	Warner Bros
Troja	1956	118	SW	Warner Bros



Titel	Jahr	StarName
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Good-Bye Lenin!	2003	Daniel Brühl
Troja	2004	Brad Pitt
Troja	1956	Stanley Baker

„Verlustbehaftete“ Zerlegung

Restaurant	Gast	Gericht
Firenze	Kowalski	Pizza
Roma	Meyer	Pizza
Roma	Kowalski	Calamari

Annahme:

Restaurant, Gast \rightarrow Gericht

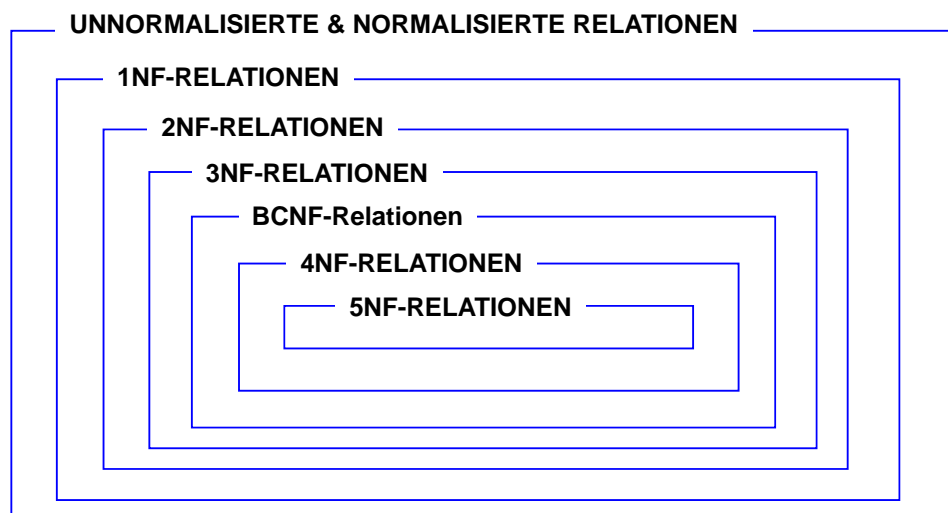
Restaurant	Gast
Firenze	Kowalski
Roma	Meyer
Roma	Kowalski

Besucht

Gast	Gericht
Kowalski	Pizza
Meyer	Pizza
Kowalski	Calamari

Isst

Normalisierung von Relationen



■ Zerlegung eines Relationenschemas \mathcal{R} in höhere Normalformen

- fortgesetzte Anwendung der Projektion im Zerlegungsprozess
- Beseitigung von Anomalien bei Änderungsoperationen
- Erhaltung aller nicht-redundanter Funktionalabhängigkeiten von \mathcal{R} (\rightarrow sie bestimmen den Informationsgehalt von \mathcal{R})
- Gewährleistung der Rekonstruktion von \mathcal{R} durch verlustfreie Verbunde
- bessere „Lesbarkeit“ der aus \mathcal{R} gewonnenen Relationen

Normalisierung von Relationen (2)

<u>PNR</u>	PNAME	FACH	STUDENT (MATNR, NAME, ...)
3678	Rahm	DBS	196481 Maier ... 123766 Coy ... 900550 Schmitt ...
1234	Brewka	KI	654711 Abel ... 123766 Coy ...

Prüfungsgeschehen

Anomalien, z. B.:

- **Insert** Student
- **Delete** Prof
- **Update** Student

■ Unnormalisierte Relation: **Non-First Normal-Form (NF²)**

- enthält "Attribute", die wiederum Relationen sind
- Darstellung von komplexen Objekten (hierarchische Sichten, Clusterbildung)

■ Nachteile:

- Unsymmetrie (nur eine Richtung der Beziehung)
- implizite Darstellung von Information
- Redundanzen bei (n:m)-Beziehungen
- Anomalien bei Aktualisierung

■ Normalisierung:

- "Herunterkopieren" von Werten führt hohen Grad an Redundanz ein → Zerlegung von Relationen
- aber: Erhaltung ihres Informationsgehaltes



Überführung in 1 NF

■ Unnormalisierte Relation

Prüfungsgeschehen (PNR, PNAME, FACH, STUDENT)

(MATNR, NAME, GEBORT, ADR, FNR, FNAME, DEKAN, PDAT, NOTE)

STUDENT = Wiederholungsgruppe mit 9 einfachen Attributen (untergeordnete Relation)

■ Normalisierung (=> 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
PRÜFUNG



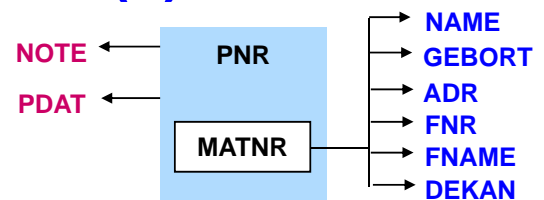
Überführung in 2NF

- 1NF verursacht immer noch viele Änderungsanomalien
 - verschiedene Entity-Mengen in einer Relation möglich bzw. Redundanz innerhalb einer Relation (Bsp.: PRÜFUNG)
- 2NF vermeidet einige Anomalien durch Eliminierung partiell abhängiger Attribute
 - Separierung verschiedener Entity-Mengen in eigene Relationen
- Def.: **Primärattribut** (Schlüsselattribut) – Attribut, das zu mind. einem Schlüsselkandidaten eines Schemas gehört.
- Ein Relationenschema \mathcal{R} ist in 2NF, wenn es
 - in 1NF ist und
 - jedes Nicht-Primärattribut von \mathcal{R} voll funktional von jedem Schlüsselkandidaten in \mathcal{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)



Überführung in 2NF (2)

- Voll funktionale Abhängigkeiten in PRÜFUNG



- Relationenschema in 2NF

Prüfung'

<u>PNR</u>	<u>MATNR</u>	PDAT	NOTE
1234	123 766	22.10.	4
1234	654 711	14.02.	3
3678	196 481	21.09.	2
3678	123 766	02.03.	4
8223	226 302	12.07.	1

Prüfer

<u>PNR</u>	PNAME	FACH
1234	Brewka	KI
3678	Rahm	DBS
8223	Weber	WI

Student'

<u>MATNR</u>	NAME	GEBORT	ADR	FNR	FNAME	DEKAN
123 766	Coy	Leipzig	XX	F11	Wirtschaftswissenschaften	A
654 711	Abel	Torgau	XY	F19	Mathematik/Informatik	B
196 481	Maier	Köln	YX	F19	Mathematik/Informatik	B
226 302	Schulz	Leipzig	YY	F11	Wirtschaftswissenschaften	A



Überführung in 3NF

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

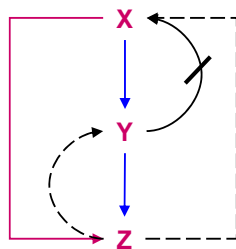
– Beispiel: Vermischung von Fakultäts- und Studentendaten in Student'

- Definitionen:

Eine Attributmeng Z von Relationenschema \mathcal{R} ist *transitiv abhängig* von einer Attributmeng X in \mathcal{R} , wenn gilt:

– X und Z sind disjunkt

– es existiert eine Attributmeng Y in \mathcal{R} , so dass gilt: $X \rightarrow Y, Y \rightarrow Z, Y \not\rightarrow X, Z \not\rightarrow Y$



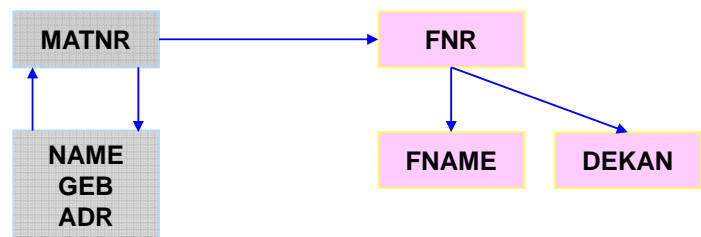
$Z \rightarrow Y$ zulässig

strikte Transitivität: $Z \not\rightarrow Y$

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

Überführung in 3NF (2)

- Funktionale Abhängigkeiten in STUDENT'



- Relationenschema in 3NF

Prüfung'

<u>PNR</u>	<u>MATNR</u>	PDAT	NOTE
1234	123 766	22.10.	4
1234	654 711	14.02.	3
3678	196 481	21.09.	2
3678	123 766	02.03.	4
8223	226 302	12.07.	1

Fakultät

<u>FNR</u>	FNAME	DEKAN
F11	Wirtschaftswissenschaften	A
F12	Medizin	C
F19	Mathematik/Informatik	B

Prüfer

<u>PNR</u>	PNAME	FACH
1234	Brewka	KI
3678	Rahm	DBS
8223	Weber	WI

Student

<u>MATNR</u>	NAME	GEBORT	ADR	FNR
123 766	Coy	Leipzig	XX	F11
654 711	Abel	Torgau	XY	F19
196 481	Maier	Köln	YX	F19
226 302	Schulz	Leipzig	YY	F11

Boyce/Codd-Normalform (BCNF)

- Definition der 3NF hat gewisse Schwächen bei Relationen mit mehreren, *sich überlappenden* Schlüsselkandidaten

- Beispiel:

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

- es bestehe eine (1:1)-Beziehung zwischen PNR und FACH
- einziges Nicht-Primärattribut: NOTE
 ⇒ PRÜFUNG ist in 3NF
- jedoch Änderungsanomalien, z. B. bei FACH

PNR	MATNR	Fach	NOTE
45	1234	Datenbanksysteme	1
45	4711	Datenbanksysteme	3
45	5678	Datenbanksysteme	2
56	1234	Künstliche Intelligenz	4

- Ziel: Beseitigung der Anomalien für Primärattribute
- Definition: Ein Attribut (oder eine Gruppe von Attributen), von dem andere voll funktional abhängen, heißt *Determinant*.
- Welches sind die Determinanten in PRÜFUNG?



Boyce/Codd-Normalform (2)

- Definition: Ein Relationenschema \mathcal{R} ist in *BCNF*, wenn es in 1NF ist und jeder Determinant ein Schlüsselkandidat von \mathcal{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: $X \rightarrow Y$ impliziert $X \rightarrow Z$

- Zerlegung von Prüfung

PRÜF (PNR, MATNR, NOTE)

FBEZ (PNR, FACH)

oder

PRÜF2 (MATNR, FACH, NOTE)

FBEZ (PNR, FACH)

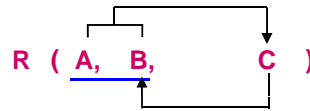
- Beide Zerlegungen führen auf BCNF-Relationen

- Änderungsanomalie ist verschwunden
- alle funktionalen Abhängigkeiten sind erhalten



Boyce/Codd-Normalform (3)

- Sind BCNF-Zerlegungen immer sinnvoll?



ist in 3NF, weil B
Primärattribut ist!

- Beispiel:
STUDENT, FACH → PRÜFER,
PRÜFER → FACH

- 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

SFP

<u>STUDENT</u>	<u>FACH</u>	<u>PRÜFER</u>
Sloppy	DBS	Sosna
Hazy	DBS	Rahm
Sloppy	KI	Brewka

- Wie sieht die BCNF-Zerlegung aus?

- Neue Probleme: STUDENT, FACH -> PRÜFER ist nun „extern“ (Konsistenzprüfung?)
- BCNF hier zu streng, um bei der Zerlegung alle funktionalen Abhängigkeiten zu bewahren (key breaking dependency)



Probleme der Normalisierung

- Weitestgehende Zerlegung nicht immer sinnvoll
- Beispiel:

Relation PERS (PNR, PLZ, ORT) mit FA PLZ → ORT

- Normalisierung verlangt Zerlegung in

- Klärungsbedarf

- Änderungshäufigkeit?
- Suchaufwand für Adresse ? (Verbundoperation) !
- Sind ORT oder PLZ in diesem Kontext eigenständige Entities (als Kandidaten für eigene Relation in 3NF)?

=> besser PERS in 2NF!



Zusammenfassung

■ Normalisierung von Relationen

- Verbesserung eines gegebenen DB-Entwurfs
- Ziel: eine Relation beschreibt nur einen Objekttyp
- Eliminierung von Änderungsanomalien
- wachsender Informationsgehalt mit zunehmender Normalisierung

■ Bestimmung aller funktionalen Abhängigkeiten

- n:1-Beziehung zwischen zwei Attributmengen einer Relation
- Wesentliche Integritätszusicherungen

■ schrittweise Normalisierung:

- 1NF: normalisierte Relationen (einfache Attribute)
- 2NF: keine partiellen (funktionalen) Abhängigkeiten
- 3NF: keine transitiven Abhängigkeiten (jedes Nicht-Primärattribut ist direkt von jedem SK abhängig)
- BCNF: jeder Determinant ist Schlüsselkandidat
- 3NF meist ausreichend

■ Überarbeitung des DB-Schemas: Stabilitätsgesichtspunkte/ Änderungshäufigkeiten können schwächere Normalformen verlangen